DEPARTAMENTO DE INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

UNA APROXIMACIÓN PARA REPRESENTAR ESTÁNDARES DE SEGURIDAD CON UNA HERRAMIENTA DE INGENIERÍA DE REQUISITOS BASADA EN ONTOLOGÍAS

Trabajo Fin de Grado

Autor: Álvaro Gómez

Tutor: José Luis de la Vara

Leganés, febrero de 2018

Agradecimientos

Muchas gracias a todas las personas que han intervenido en este Trabajo de Fin de Grado. En particular, agradezco a mi tutor sus consejos y orientación que han hecho posible la realización de este trabajo y a mi familia por estar siempre a mi lado. Gracias.

ÍNDICE

Contenido

ÍNDICE DE FIGURAS:

ÍNDICE DE TABLAS

RESUMEN EJECUTIVO:

Los sistemas críticos de seguridad son aquellos sistemas cuyo fallo puede ocasionar pérdidas de vidas, daños materiales significativos o daños al medio ambiente.

Los sistemas críticos deben cumplir con normas de seguridad y estándares de seguridad como una forma de garantizar que no pueden provocar riesgos indebidos para las personas, la propiedad o el medio ambiente. Un estándar de seguridad ('safety standard') es un documento que recoge un conjunto de buenas prácticas, acordadas por un consorcio de empresas y profesionales, para el desarrollo y aseguramiento de sistemas críticos de seguridad.

El cumplimiento de las normas de seguridad es una actividad muy exigente, ya que los estándares pueden constar de cientos de páginas y los profesionales generalmente tienen que demostrar el cumplimiento de miles de criterios relacionados con la seguridad.

Estos documentos suelen ser largos, ambiguos, y difíciles de entender, por lo que varios expertos recomiendan su representación explícita y estructurada para facilitar la comprensión y aplicación de estos estándares.

Dado que la realización de estas representaciones puede ser compleja, es aconsejable utilizar herramientas que la apoyen.

El objetivo de este TFG es definir una aproximación para representar estándares de seguridad en KM, una herramienta de ingeniería de requisitos basada en ontologías que se utiliza actualmente en industria para representar, por ejemplo, los requisitos y la estructura de sistemas.

La aproximación utilizará además como base las propuestas existentes más recientes para el modelado de estándares de seguridad.

1.- INTRODUCCIÓN:

En este primer apartado se presenta el problema a tratar, los objetivo que se pretenden conseguir y la estructura del documento.

Para ello, en primer lugar, se sitúa el problema en su contexto y se explican cuáles han sido las motivaciones a través de las cuales se ha visto oportuno analizar y buscar solución a dicho problema.

Una vez situado y explicadas las motivaciones, se definirán los objetivos propuestos. Por último, se detallará la estructura del documento y las abreviaturas y acrónimos.

1.1.- Contexto:

Los sistemas críticos de seguridad son aquellos sistemas cuyo fallo puede ocasionar pérdidas de vidas, daños materiales significativos o daños al medio ambiente. Hay muchos ejemplos bien conocidos en áreas de aplicación tales como dispositivos médicos, control de vuelo de aeronaves, armas y sistemas nucleares. (Knight, 2015).

Muchos sistemas de información modernos se están convirtiendo en críticos de seguridad en un sentido general, esto es debido a las posibles pérdidas financieras e incluso de vidas que pueden resultar de sus fallos. (*Ibídem*).

Desde una perspectiva de software, el desarrollo de sistemas críticos de seguridad en los números requeridos y con una fiabilidad adecuada va a requerir avances significativos en áreas tales como especificación, arquitectura, verificación y proceso. (*Ibídem*).

Los problemas visibles que han surgido en el ámbito de la seguridad de los sistemas de información sugieren que la seguridad es también un desafío importante. (*Ibídem*).

La certificación es un requisito previo importante para la mayoría de los sistemas críticos de seguridad antes de que puedan ser puestos en funcionamiento. (Panesar-Walawege, Sabetzadeh y Briand, 2011).

Durante la certificación, los proveedores de sistemas críticos de seguridad a menudo tienen que presentar un conjunto coherente de evidencias que demuestran que los sistemas desarrollados son seguros para la operación. (*Ibídem*).

Independientemente del enfoque de certificación adoptado (basado en el proceso o en el producto), la recopilación de evidencias apropiadas durante el desarrollo es crítica para la certificación exitosa.

En la actualidad, tanto los proveedores de sistemas críticos de seguridad como los organismos de certificación se enfrentan a varios desafíos en relación con la recopilación y cumplimiento de pruebas de seguridad. (*Ibídem*).

En particular, les cuesta interpretar los requisitos de evidencias impuestos por las normas de seguridad dentro del ámbito de aplicación. Existe poca ayuda para registrar, consultar e informar las evidencias de una manera estructurada. Y hay una ausencia general de directrices sobre cómo las evidencia recogidas apoyan los objetivos de seguridad. *(Ibídem).*

Este tipo de problemas se está abordando en AMASS (Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems; https://www.amass-ecsel.eu/), un proyecto de investigación del programa H2020 que tiene por fin crear una plataforma de herramientas abiertas, un ecosistema, y una comunidad auto sostenible a nivel europeo para asegurar y certificar Sistemas Ciber-Físicos en los mercados industriales verticales más grandes, incluyendo automoción, ferrocarril, aeroespacial, espacio, energía. (*Ibídem*).

1.2.- Motivación:

La mayoría de los sistemas críticos de seguridad deben cumplir con las normas de seguridad como una forma de asegurar que no pueden provocar riesgos indebidos. Ejemplos de estas incluyen IEC 61508 para una amplia gama de industrias, DO-178C en aviónica, EN 50128 en el sector del ferrocarril, y ISO 26262 en la industria automotriz. (José Luis de la Vara, Álvaro Gómez, Elena Gallego, Gonzalo Génova y Anabel Fraga, 2017).

Los estándares de seguridad son típicamente documentos textuales grandes que consisten en cientos de páginas y definen miles de criterios para el cumplimiento. (*Ibídem*).

La complejidad resultante puede dificultar la comprensión de un estándar. La ambigüedad y contradicciones también son habituales en su texto, y los proveedores del sector han reconocido, problemas en la comprensión y aplicación de las normas. (*Ibídem*).

Esto puede conducir a riesgos de certificación, ya que un proveedor del sistema puede fallar o malinterpretar algunos criterios y por lo tanto no desarrollar un sistema compatible. (*Ibídem*).

Como solución, este TFG propone el uso de representaciones estructuradas de las normas de seguridad con el objetivo de ayudar a los proveedores de sistemas críticos a comprender y seguirlas. (*Ibídem*).

Estas representaciones se han realizado, más a menudo, en los modelos basados en UML, como un diagrama de clases o un perfil UML. Sin embargo, las representaciones también pueden desarrollarse con tecnologías semánticas, por ejemplo, como una

ontología que incluye los conceptos principales y las relaciones entre los conceptos de una norma de seguridad. (*Ibídem*).

Varios autores abogan por la representación explícita de los estándares con modelos, que pueden ser creados con tecnologías semánticas como ontologías. Sin embargo, esta posibilidad ha recibido poca atención. (*Ibídem*).

Además, los pocos autores que lo han abordado también sólo han tratado un subconjunto de aspectos de los estándares de seguridad y han utilizado tecnologías que no suelen aplicarse para la ingeniería de sistemas críticos.

A continuación, se hará un resumen de los avances realizados en este campo:

- Gallina y Szatmári proponen la creación de modelos basados en ontologías para facilitar la comparación de las normas de seguridad. Representan las normas ISO 26262 y EN 50128 con OWL 2.0 y Protegé para posteriormente generar líneas de productos orientadas a la seguridad en SPEM. La ontología se centra en las actividades de las normas. (Gallina y Szatmári, 2015).

- Jost et. al. proponer la formalización de la norma ISO 26262 con una ontología para permitir la selección semi-automatizado de los requisitos de la norma. De esta manera, la ISO 26262 puede adaptarse a un proyecto dado. Jost et al. Combinar OWL y SPEM, y gestionar la ontología con Protegé y Pellet, centrándose en la terminología estándar. (Jost, 2015).

- Luo et al. proponer un enfoque basado en modelos para el cumplimiento de las normas de seguridad y para facilitar la reutilización de aseguramiento. Utilizan Protegé y OWLGrEd para especificar y visualizar, respectivamente, modelos conceptuales de estándares de seguridad, y combinarlos con UML y SPEM. El enfoque se aplica a ISO 26262, y la ontología se centra en la terminología del estándar. (Luo, 2013).

Se encuentran tres debilidades principales en el estado del arte:

- En primer lugar, se ha prestado poca atención al uso de tecnologías semánticas para representar las normas de seguridad, por lo que sus beneficios (por ejemplo, el razonamiento automático) apenas se han estudiado.

- En segundo lugar, las tecnologías propuestas se han centrado en aspectos específicos de las normas, en concreto, en sus actividades y su terminología. Sin embargo, el cumplimiento requiere la consideración de más aspectos, por ejemplo, los artefactos a gestionar y las relaciones entre ellos. Por lo tanto, no se ha desarrollado aún ninguna propuesta que proporcione una representación ontológica integrada.

- En tercer lugar, las tecnologías semánticas adoptadas en la literatura rara vez o nunca se utilizan en la industria para la ingeniería de sistemas críticos, lo que da lugar a una brecha entre la investigación y la práctica. No tenemos conocimiento de ninguna empresa que utilice OWL o Protege en proyectos reales, y los estudios relacionados sobre el estado de la práctica no proporcionan evidencia de su uso.

Este trabajo pretende hacer frente a estos problemas mediante la investigación de cómo la herramienta KM (Knowledge Manager) se puede utilizar para representar a los estándares de seguridad y posteriormente explotar la representación resultante.

KM es una herramienta utilizada en entornos industriales para la ingeniería de sistemas críticos para representar el conocimiento del dominio con ontologías.

Estas ontologías abarcan varios aspectos, desde la terminología del sistema hasta los patrones de especificación del sistema, y pueden usarse para diferentes propósitos, por ejemplo, especificación del sistema, análisis de la calidad del artefacto del sistema y reutilización de la información del sistema.

El uso de KM en la práctica se centra en características específicas del sistema, por ejemplo, estructura del sistema, pero argumentamos que tal uso puede extenderse para apoyar el cumplimiento de las normas de seguridad.

1.3.- Objetivo:

El objetivo de este TFG consiste en **definir una aproximación para representar estándares de seguridad en KM**, una herramienta de ingeniería de requisitos, perteneciente al conjunto de herramientas de RQS, basada en ontologías que se utiliza actualmente en industria para representar, por ejemplo, los requisitos y la estructura de sistemas. La aproximación utilizará además las propuestas existentes más recientes para el modelado de estándares de seguridad.

De esta manera se conseguirá que el análisis de los estándares de seguridad para su posterior uso en la creación de sistemas críticos de seguridad se convierta en una tarea más simple y no tan compleja como lo es hoy en día.

Dentro de este objetivo general se pueden diferenciar los siguientes objetivos específicos:

- Determinar la correspondencia entre el contenido de una estándar de seguridad y una ontología con la herramienta KM.
    - Este objetivo requerirá a su vez:
        - Comprensión de los estándares.
        - Análisis y representación de la ontología de un estándar.
        - Lograr manejar a la perfección el paquete de herramientas RQS.
- Aplicar y validar la aproximación en casos reales.

1.4.- Estructura del documento:

La estructura del documento consta de las siguientes partes:

En el capítulo 1, se realiza una introducción en la que se presenta el tema, se sitúa en su contexto y se enuncian los objetivos propuestos por este trabajo.

En el capítulo 2, se realiza una explicación de la terminología del tema, se sitúa en su entorno socioeconómico y en su marco legal. También en este capítulo, se presentan las herramientas a utilizar y el trabajo relacionado.

Una vez enunciado y presentado el tema, en el capítulo 3, se realiza la propuesta de aproximación, se presenta el metamodelo a partir del cual se va a basar la aproximación y se explica la forma en la que llevarla a cabo.

En el capítulo 4, se realiza la aproximación citada en el capítulo anterior sobre dos casos reales. En este capítulo se corroborará la aplicación y validación de la aproximación propuesta.

Para finalizar, después de exponer y justificar la a aproximación propuesta, se exponen las conclusiones y las líneas futuras propias del autor del trabajo, capítulo 5.

El trabajo también dispone de un Anexo, en el que se encuentran el diagrama de Gantt asociado al proyecto y el presupuesto.

1.5.- Abreviaturas y acrónimos:

KM	Knowledge Manager
DO-178C	Normas que rigen el sector aeroespacial
EN 50128	Normas que rigen el sector ferroviario
RQS	Requisitos Calidad Suite
RQA	Analizador de Calidad de Requisitos
RAF	Reference Assurance Framework

2.- ANTECEDENTES:

En este apartado se analiza el contexto vinculante en el que se encuentra el tema propuesto por este TFG.

En primer lugar, se sitúa el tema en su entorno socioeconómico y al marco regulador al que pertenece. Después se realiza una introducción de la terminología de los sistemas críticos de seguridad, su modo de certificación y sus estándares.

Por último, se analizan las herramientas y trabajo relacionados los cuáles han servido de base para el trabajo desarrollado.

2.1.- Datos económicos del mercado:

Un moderno marcapasos cardíaco es un ordenador con periféricos especializados, el combatiente F22 de la Fuerza Aérea de los Estados Unidos depende en gran medida de una red informática, al igual que un coche moderno, y muchas instalaciones de defensa son en realidad sistemas informáticos distribuidos. Estos y muchos otros sistemas son ejemplos de los llamados sistemas de seguridad crítica. (Knight, 2015).

Muchos sistemas modernos dependen de ordenadores para su correcto funcionamiento. Por supuesto, la mayor preocupación son los sistemas críticos para la seguridad porque sus consecuencias de su fallo pueden ser considerables.

Existen muchas aplicaciones que tradicionalmente han sido consideradas críticas para la seguridad, pero el alcance de la definición tiene que ser ampliado a medida que los sistemas informáticos continúan siendo introducidos en muchas áreas que afectan nuestras vidas. Es probable que en un futuro se incremente drásticamente el número de sistemas informáticos que consideramos críticos para la seguridad. (*Ibídem*).

2.1.1.- Ejemplos prácticos:

A continuación, se citarán una serie de ejemplos reales con el fin de obtener una mejor comprensión del entorno socioeconómico sobre el cual gira el trabajo:

Un ejemplo de lo que puede salir mal con los sistemas de apoyo de diseño se produjo con dos programas que realizan análisis de elementos finitos. Estos programas se utilizan ampliamente en el diseño de ingeniería, en particular el diseño estructural. (Knight, 2015).

En mayo de 1996, la Comisión de Reglamentación Nuclear publicó el Aviso de Información NRC 96-29 dirigido a todos los titulares de licencias de explotación o permisos de construcción para reactores nucleares. La Comisión había determinado que en ese momento se habían notificado 150 errores para estos programas, pero la Comisión

no sabía cómo se habían utilizado los programas en el análisis de la seguridad en las centrales nucleares. Se pidió a los destinatarios de la notificación que revisaran la información para determinar su aplicabilidad. Las implicaciones son obvias. (*Ibídem*).

Un tipo diferente de problema surgió con el sistema de protección primaria para el reactor de energía nuclear Sizewell B en el Reino Unido. Este sistema fue implementado en software y fue necesario para lograr una fiabilidad de no más de 10^{-4} fallos por demanda. (*Ibídem*).

Una vez completado el diseño del sistema, utilizó más de 650 microprocesadores y 1.200 tarjetas de circuitos, y el software tenía más de 70.000 líneas. Cuando se realizaron pruebas del sistema, el sistema sólo pasó el 48% de los casos de prueba.

Se informó que el sistema había fallado al otro 52%, pero, de hecho, el problema real era que, para muchas de las pruebas, no era posible determinar si la prueba había sido pasada. Este sistema fue diseñado para cerrar el reactor cuando surgía algún tipo de problema, una aplicación de seguridad crítica. Alcanzar el objetivo de confiabilidad parece improbable para un sistema con muchos ordenadores, este número de líneas de código y este historial de pruebas. (*Ibídem*).

El 26 de octubre de 1992, el servicio de ambulancias de la ciudad de London, Inglaterra, pasó de un sistema de despacho manual a un sistema de despacho asistido por ordenador.

La conmutación se realizó de una vez, de modo que se esperaba que el sistema computarizado funcionara para toda la zona de cobertura. El sistema funcionó inicialmente, pero una sucesión compleja de eventos llevó al sistema esencialmente a no ser operativo ya que la demanda aumentó durante el día. (*Ibídem*).

Dado que el despacho se vio gravemente retrasado en muchos casos, hay buenas razones para pensar que muertes o lesiones fueron fruto de este fracaso. (*Ibídem*).

Hay muchas lecciones que se pueden aprender de incidentes como estos, pero, por desgracia, las lecciones a veces se pierden. El fracaso de Mars Climate Orbiter, por ejemplo, se produjo porque el sistema equivocado de unidades se utilizó en parte del software de tierra. (*Ibídem*).

2.2.- Marco legal:

En este punto se estudia el marco legal definido para este trabajo. Para ello se realiza una distinción entre las normas generales y específicas. Las normas llevadas a cabo pertenecen a la UE.

- Normas generales: Las normas generales por las que se encuentra regido el marco legal de los sistemas críticos de seguridad son las siguientes:

- Ley Oficial de Protección de Datos (LOPD): La presente Ley Orgánica tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar (BOE, 1999).

- Ley de Propiedad Industrial**:** Conjunto de derechos que pueden tener las personas, bien sean físicas o jurídicas, sobre una invención, una marca o un diseño industrial. Da a la persona el derecho de utilizarla y el derecho a prohibir que lo haga una tercera persona (BOE, 1999).

- Normas específicas: Las normas específicas y los órganos certificadores dependiendo del ámbito del sistema crítico de seguridad son:

  - CENELEC: Conjunto de normas que rigen el sector ferroviario. La encargada de su publicación es la entidad ERA (*European Railway Agency*).

  - ECSS: Conjunto de normas que rigen el sector aeroespacial. La encargada de su publicación es la entidad ESA (*European Sépase Agency*).

  - DO-178 y ARP: Conjunto de normas que rigen el sector aeroespacial. La encargada de su publicación es la entidad EASA (*European Aviation Safety Agency*).

  - CE: Conjunto de normas que rigen el sector de dispositivos médicos.

  - IEC61508: utilizado para la evaluación de certificación de sistemas relacionados con la seguridad en el ámbito electrónico o programable.

  - ISO-26262: Conjunto de normas que rigen el sector de automovilismo.

2.3.- Sistemas críticos de seguridad:

Hay un montón de definiciones de *sistema de seguridad crítico*, pero la noción intuitiva en realidad funciona bastante bien. La preocupación tanto de forma intuitiva y formalmente es con *el offailure consecuencias.* (Knight, 2015).

Si el fallo de un sistema puede dar lugar a consecuencias que se determinan como inaceptables, entonces el sistema es crítico para la seguridad. En esencia, un sistema es crítico para la seguridad cuando dependemos de él para nuestro bienestar. En esta sección, las implicaciones de esta idea se exploran en términos de las clases de sistemas que deben considerarse críticos para la seguridad. (*Ibídem*).

2.3.1.- Sistemas Tradicionales

Las áreas tradicionales que se han considerado el hogar de los sistemas críticos de la seguridad incluyen la atención médica, los aviones comerciales, la energía nuclear, y las armas. El fracaso en estas áreas puede conducir rápidamente a la vida humana se pone en peligro, la pérdida de equipo, y así sucesivamente. (Knight, 2015).

Los ordenadores se utilizan en la medicina mucho más ampliamente que la mayoría de las personas se dan cuenta. La idea de usar un microprocesador para controlar una bomba de insulina es bastante conocida. El hecho de que un marcapasos sea en gran medida un ordenador es menos conocido. (*Ibídem*).

El uso extensivo de ordenadores en procedimientos quirúrgicos es casi desconocido excepto por especialistas. Equipos "computerized" están haciendo incursiones en procedimientos tales como el reemplazo de cadera, la cirugía espinal y la cirugía oftálmica.

En los tres casos, los dispositivos robóticos controlados por computadora están reemplazando a las herramientas tradicionales de los cirujanos y proporcionando beneficios sustanciales a los pacientes. (*Ibídem*).

El Boeing 777 es descrito por Boeing como "el avión tecnológicamente más avanzado del mundo". Muchas tecnologías diferentes han contribuido a la aeronave incluyendo los sistemas informáticos críticos para la seguridad. Hay seis principales pantallas planas y varias otras pantallas más pequeñas en la cabina. (*Ibídem*).

La aeronave cuenta con varios sistemas informatizados importantes para ayudar al piloto, incluyendo la gestión de vuelo y una mayor alerta de proximidad al suelo. Gran parte de los equipos mecánicos e hidráulicos tradicionales se obvia mediante el uso de un sistema de control de vuelo por cable. (*Ibídem*).

El sistema de control de vuelo primario Boeing 777 utiliza tres canales separados para la redundancia. Cada canal se implementa con tres carriles separados, cada uno de los cuales utiliza procesadores diferentes y compiladores diferentes. La red extensa proporciona la comunicación necesaria entre los diferentes subsistemas. (*Ibídem*).

2.3.2.- Sistemas no tradicionales:

El alcance del concepto de sistema crítico para la seguridad es amplio y esa amplitud debe tenerse en cuenta cuando los profesionales y los investigadores tratan con sistemas específicos. (Knight, 2015).

Un examen más detallado del tema revela que muchos nuevos tipos de sistemas tienen el potencial de consecuencias muy altas de falla, y estos sistemas también deberían

considerarse críticos para la seguridad. Es obvio que la pérdida de un avión comercial probablemente matará a la gente. (*Ibídem*).

No es obvio que la pérdida de un sistema telefónico podría matar a la gente. Sin embargo, una pérdida prolongada del servicio 911 sin duda resultará en lesiones graves o la muerte. (*Ibídem*).

El servicio Emergencia 911 es un ejemplo de una aplicación de infraestructura crítica. Otros ejemplos son el control del transporte, los sistemas bancarios y financieros, la generación y distribución de electricidad, las telecomunicaciones y la gestión de los sistemas de agua. (*Ibídem*).

Todas estas aplicaciones están ampliamente informatizadas y el fallo de la computadora puede conducir, y, de hecho, a una pérdida extensiva de servicios con la consecuente interrupción de las actividades normales. En algunos casos, la interrupción puede ser muy grave. La pérdida generalizada de suministro de agua o electricidad tiene implicaciones obvias para la salud y la seguridad. (*Ibídem*).

Del mismo modo, la pérdida generalizada de servicios de transporte, como el transporte por ferrocarril y camiones, afectaría a la distribución de alimentos y energía. Es prudente poner los sistemas informáticos de los que dependen las infraestructuras críticas en la categoría de seguridad crítica. (*Ibídem*).

2.4.- Certificación de sistemas críticos de seguridad:

La mayoría de los sistemas críticos que utilizan sistemas informáticos en ámbitos como el aeroespacial, el ferrocarril y el automóvil están sujetos a alguna forma de evaluación de la seguridad por parte de un tercero (por ejemplo, una autoridad de certificación) como forma de asegurar que no plantean riesgos indebidos para las personas, la propiedad o el medio ambiente. (José Luis de la Vara, Alejandra Ruiz, Katrina Attwood, Huáscar Espinoza, Rajwinder Naur Panesar-Walawege, Ángel López, Idoya del Río, Tim Kelly, 2016).

Un tipo común de evaluación es el cumplimiento de las normas de seguridad (o relacionadas con la seguridad), usualmente denominadas certificación de seguridad. (*Ibídem*).

Ejemplos de normas de seguridad utilizadas en la industria incluyen IEC 61508 para sistemas electrónicos eléctricos, electrónicos y programables en una amplia gama de industrias y normas más específicas como la DO-178C para la aviónica, las normas CEN-ELEC para el ferrocarril (Por ejemplo, EN 50128), e ISO 26262 para el sector del automóvil. (*Ibídem*).

La demostración del cumplimiento con las normas de seguridad suele ser costoso y lleva mucho tiempo, y puede ser muy difícil. (*Ibídem*).

En primer lugar, los proveedores de sistemas tienen que recopilar evidencias de cumplimiento tales como análisis de peligros, resultados de pruebas y registros de actividad para demostrar que se han cumplido los criterios de seguridad de una norma. (*Ibídem*).

Para recolectar estas evidencias, los practicantes deben determinar los objetivos de seguridad a alcanzar y el proceso a ejecutar en base a las características de un sistema particular. (*Ibídem*).

Como el texto de las normas de seguridad puede ser ambiguo, inconsistente y difícil de entender, esto puede convertirse en una ardua tarea. En segundo lugar, los profesionales generalmente tienen que manejar grandes cantidades de evidencia para mostrar cómo un sistema cumple con un estándar. Si la evidencia no está estructurada adecuadamente, su volumen y complejidad pueden poner en peligro la certificación de seguridad. (*Ibídem*).

La demostración del cumplimiento de las normas de seguridad se vuelve aún más difícil cuando un sistema cambia. Por ejemplo, la evidencia evoluciona cuando un sistema pretende ser certificado con diferentes estándares de seguridad o reutilizado en otro dominio de aplicación. Estas son actualmente importantes preocupaciones en la industria, y la mayoría de los profesionales han enfrentado estas situaciones. Aunque la correspondencia entre las normas de seguridad ha comenzado a estudiarse, es una tarea compleja. (*Ibídem*).

Normalmente no existe una combinación perfecta entre las necesidades de cumplimiento de diferentes normas de seguridad, y los proveedores de sistemas suelen tener sus propias interpretaciones y por lo tanto el uso de un estándar. Como resultado, el cumplimiento de una nueva norma nunca es sencillo. La industria necesita medios que permitan la reutilización de las pruebas y apoyen el análisis del impacto del cambio de las pruebas en general, y en las situaciones entre dominios y entre normas, en particular. (*Ibídem*).

Todos los desafíos anteriores pueden llevar a riesgos de certificación, ya que un proveedor del sistema podría no ser capaz de desarrollar un sistema seguro, demostrar el cumplimiento de una norma de seguridad o ayudar a un tercero a ganar confiabilidad en la seguridad del sistema. Abogamos por el uso de enfoques basados en modelos para abordar estos desafíos. (*Ibídem*).

Los modelos pueden facilitar la comprensión de las normas de seguridad, la identificación de incoherencias en su texto, la determinación de las pruebas a recoger, la especificación de los requisitos de trazabilidad y la evaluación del cumplimiento. (*Ibídem*).

2.5.- Estándares para sistemas críticos de seguridad:

Se entiende por norma o estándar al conjunto de requisitos que debe cumplir un determinado sistema independientemente del sector al que se desarrolle (Nair, 2014).

Las normas de certificación que se utilizan para el proceso de certificación pueden ser nacionales o internacionales, y genéricas (aplicadas a un sector específico) o específicas (aplicadas a un tipo de sistema concreto dentro de ese sector). (*Ibídem*).

Algunos ejemplos de estándares de seguridad para sistemas críticos son (ya citados en el punto 2.2):

- CENELEC: Conjunto de normas que rigen el sector ferroviario. La encargada de su publicación es la entidad ERA (*European Railway Agency*).

- ECSS: Conjunto de normas que rigen el sector aeroespacial. La encargada de su publicación es la entidad ESA (*European Space Agency*).

- DO-178 y ARP: Conjunto de normas que rigen el sector aeroespacial. La encargada de su publicación es la entidad EASA (*European Aviation Safety Agency*).

- CE: Conjunto de normas que rigen el sector de dispositivos médicos.

- IEC61508: utilizado para la evaluación de certificación de sistemas relacionados con la seguridad en el ámbito electrónico o programable.

- ISO-26262: Conjunto de normas que rigen el sector de automovilismo.

Por lo general, un sistema específico puede ajustarse a varios estándares, y dependerá del cliente o del contrato que se tenga con él cuál es el que se va a seguir. El uso o la certificación del sistema por parte de un estándar da al usuario la seguridad de que su sistema ha sido probado y certificado por un órgano regulador. *(Ibidem).*

Este certificado se puede usar también en caso de fallos, dejando constancia de que, aunque se haya producido un error, el sistema cumplía con la normativa reguladora del mismo. *(Ibídem).*

2.6.- RQS:

Requisitos Calidad Suite (RQS) es un conjunto de herramientas que soportan la planificación, personalización, medición, control y gestión de sus productos de trabajo, especialmente de sus especificaciones de requisitos, mejorando así su calidad. (The Reuse Company, 2016).

RQS evalúa cuán estrechamente sus requisitos coinciden con las características de calidad descritas en referencias conocidas como IEEE Std. 830, IEEE 29148 o el ESA PSS-O5 y otros. (*Ibídem*).

Estas características se evalúan por medio de un conjunto objetivo y fácil de medir (también conocido como reglas o indicadores) que se pueden personalizar fácilmente al nivel de madurez de sus procesos organizativos o proyecto y que siempre se corresponde con el conjunto de directrices o listas de verificación definidas y mandato de su organización. (*Ibídem*).

RQS puede analizar los siguientes idiomas: inglés, francés, alemán, sueco y español, pero se puede adaptar fácilmente para cualquier otro idioma cuando sea necesario. (*Ibídem*).

Herramientas incluidas en la suite:

- **Analizador de Calidad de Requisitos (RQA):** permite al proyecto medir la calidad de sus requerimientos con las guías y listas de verificación de la organización. También ayuda al equipo de Control de Calidad (QC) durante las actividades de verificación para evaluar el cumplimiento.

- **Herramienta de creación de requisitos (RAT):** ayuda al equipo encargado de redactar las especificaciones de los requisitos durante esta actividad crítica: comprobación de la exactitud, comprobación de la coherencia, reutilización de los requisitos ... todo al vuelo y en tiempo real. Además, la asistencia de escritura se basa en patrones (a.k.a. boilerplates o plantillas de nivel de declaración) que promueve la consistencia y mejora la reutilización.

- **Knowledge Manager (KM)**: KM es una herramienta utilizada en entornos industriales para la ingeniería de sistemas críticos para representar el conocimiento del dominio con ontologías.

  Estas ontologías abarcan varios aspectos, desde la terminología del sistema hasta los patrones de especificación del sistema, y pueden usarse para diferentes propósitos, por ejemplo, especificación del sistema, análisis de la calidad del artefacto del sistema y reutilización de la información del sistema.

El uso de KM en la práctica se centra en características específicas del sistema, por ejemplo, estructura del sistema, pero argumentamos que tal uso puede extenderse para apoyar el cumplimiento de las normas de seguridad.

- **RQS (Requirements Quality Suite) Server:** Este servidor se puede conectar a archivos Excel, XML, o DOORS de IBM; próximamente también lo hará con archivos Word.

La herramienta RQS fue creada por la compañía REUSE para manejar de forma eficiente la calidad de los requisitos. A parte de los productos explicados arriba, la aplicación incluye una serie de herramientas además de definir un ciclo de PDCA para proporcionar la metodología necesaria para el proceso. (*Ibídem*).

2.7.- Trabajo relacionado:

En este apartado se presenta el trabajo relacionado. Los primeros dos trabajos han servido a modo de base para darnos una visión general sobre el tema a tratar y los restantes abordan dicho tema de manera más específica.

1. ***Certifiably Safe Software-Dependent Systems: Challenges and Directions.***
   Autores: John Hatcliff, Alan Wassyng, Tim Kelly, Cyrille Comar, Paul Jones.

   Este artículo analiza el futuro de la ingeniería de software con el objetivo de apoyar el desarrollo y la certificación de sistemas dependientes de software críticos para la seguridad.

   La cantidad y el impacto de la dependencia del software en los sistemas críticos que inciden en la vida cotidiana está aumentando rápidamente. En muchos de estos sistemas, la ingeniería inadecuada de software e ingeniería de sistemas puede conducir a desastres económicos, lesiones o muerte.

   La sociedad generalmente no reconoce el potencial de pérdidas por deficiencias de sistemas debido al software hasta después de que ocurra algún contratiempo.

   Por un lado, hay aumentos sin precedentes, exponenciales en tamaño, interdependencias, complejidades, números y variedad en los sistemas y distribución de procesos de desarrollo a través de organizaciones y culturas.

   Por otro lado, la capacidad de la industria para verificar y validar estos sistemas no se ha mantenido.

El mero cumplimiento de las normas, técnicas y regulaciones existentes no puede garantizar las propiedades de seguridad de estos sistemas. La brecha entre la práctica y la capacidad está aumentando rápidamente.

Los obstáculos más importantes para el cierre de estas brechas, que propone este artículo son los siguientes:

- La educación parece estar en el centro de muchos de nuestros problemas. Casi no hay cursos de ingeniería de software o computadora ciencias que se ocupan de los temas esenciales en la construcción y / o certificación de sistemas de seguridad críticos.

- La insuficiencia de los requisitos en la práctica común. La obtención de requisitos es una actividad muy compleja. La documentación de los requisitos para que sean completos, sin ambigüedades, correctos y comprensibles es una tarea bastante compleja y a la que no se le dedica el tiempo y la importancia que tiene.

- La complejidad de los sistemas que necesitamos para construir, así como la creciente tendencia hacia el desarrollo dirigido por modelos, ha cambiado de manera fundamental la forma de certificación.

2. ***Safety Critical Systems: Challenges and Directions.*** Autores: John C. Knight.

En este artículo se realiza un análisis de la terminología, del entorno y los desafíos futuros de los sistemas críticos.

Los sistemas críticos son aquellos sistemas cuyo fallo puede ocasionar pérdidas de vidas, daños materiales significativos o daños al medio ambiente.

Muchos sistemas de información modernos se están convirtiendo en críticos de seguridad en un sentido general, porque la pérdida financiera y la pérdida de la vida puede resultar de su fracaso. Los sistemas de seguridad críticos futuros serán más comunes y más potentes.

Desde una perspectiva de software, el desarrollo de sistemas críticos de seguridad en los números requeridos y con una fiabilidad adecuada va a requerir avances significativos en áreas tales como especificación, arquitectura, verificación y proceso.

Los problemas visibles que han surgido en el ámbito de la seguridad del sistema de información sugieren que la seguridad es también un desafío importante.

Existen muchas definiciones de sistema de seguridad, pero la noción intuitiva en realidad funciona bastante bien.

La preocupación de forma intuitiva y formalmente es con el offailure consecuencias. Si el fallo de un sistema puede dar lugar a consecuencias que se determinan como inaceptables, entonces el sistema es crítico para la seguridad.

En esencia, un sistema es crítico para la seguridad cuando dependemos de él para nuestro bienestar. En esta sección, las implicaciones de esta idea se exploran en términos de las clases de sistemas que deben considerarse críticos para la seguridad.

Los desafíos a los que los sistemas críticos se enfrentarán en un futuro serán los siguientes:

- El número de sistemas interactivos de seguridad crítica presentes en una sola aplicación obligará a compartir recursos entre sistemas. Esto eliminará un elemento arquitectónico importante que da confianza en el funcionamiento correcto - la separación física.

- Las interrupciones en la interacción entre la ingeniería de software y la ingeniería de sistemas siguen siendo una causa importante de fallas. Es esencial que se desarrollen enfoques globales para modelar el sistema total de modo que se puedan analizar propiedades de sistemas enteros. Tales enfoques deben acomodar adecuadamente el software y proporcionar modelos de alta fidelidad de las características críticas del software.

- Las especificaciones defectuosas del software están implicadas en muchos fallos serios, y está claro que tenemos dificultad para indicar exactamente qué software se requiere hacer. Hay muchos aspectos de especificación que no están soportados por ninguna técnica actual, e incluso cuando existen técnicas de especificación, sigue habiendo una falta de integración para permitir el análisis completo de la especificación.

- La verificación por pruebas es imposible para sistemas que tienen que operar en lo que se ha llamado el rango ultra-confiable. Sin embargo, en la práctica, hay pocas opciones. La verificación formal y la comprobación de modelos son tecnologías deseables, pero están limitadas en su aplicabilidad. Para lograr la confianza en la gran variedad de sistemas críticos para la seguridad que se esperan, será esencial contar con enfoques de verificación de alto rendimiento, rápidos y completos.

- El tiempo y el esfuerzo de desarrollo de los sistemas críticos para la seguridad son tan extremos con la tecnología actual que la construcción de los sistemas que se exigirán en el futuro no será posible en muchos casos.

La seguridad se está convirtiendo en un tema cada vez más importante en el campo de los sistemas críticos para la seguridad, y debe ser abordada de manera exhaustiva si se van a operar con éxito los sistemas de seguridad crítica.

El desafío aquí radica mucho en el campo de la ingeniería de software en lugar de la tecnología de seguridad.

3. ***Using Model-Driven Engineering for Managing Safety Evidence: Challenges, Vision and Experience.*** Autores: Rajwinder Kaur Panesar-Walawege, Mehrdad Sabetzadeh, Lionel Briand.

En este artículo, se tratan los desafíos a los que se enfrentan los proveedores de sistemas y certificadores al tener que certificar los sistemas con los estándares de seguridad.

La certificación es un requisito previo importante para la mayoría de los sistemas críticos de seguridad antes de que puedan ser puestos en funcionamiento.

Durante la certificación, los proveedores de sistemas a menudo tienen que presentar un conjunto coherente de pruebas que demuestran que los sistemas desarrollados son seguros para la operación.

Independientemente del enfoque de certificación adoptado (basado en el proceso o en el producto), la recopilación de evidencia apropiada en la etapa adecuada de desarrollo es crítica para la certificación exitosa.

En la actualidad, tanto los proveedores de sistemas como los organismos de certificación se enfrentan a varios desafíos en relación con la recopilación de pruebas de seguridad.

En particular, les cuesta interpretar los requisitos de evidencia impuestos por las normas de seguridad dentro del ámbito de aplicación; Existe poca ayuda para registrar, consultar e informar la evidencia de una manera estructurada; Y hay una ausencia general de directrices sobre cómo la evidencia recogida apoya los objetivos de seguridad. Por lo que hace complicada la tarea de formular pruebas de seguridad.

Como solución en este artículo se propone la aplicación de la ingeniería dirigida por modelos como un facilitador para realizar las diversas tareas relacionadas con la gestión de pruebas de seguridad.

4. ***Model-based specification of safety compliance needs for critical systems: A holistic generic metamodel.*** Autores: José Luis de la Vara, Alejandra Ruiz, Katrina Attwood, Huáscar Espinoza, Rajwinder Kaur Panesar-Walawege, Ángel López, Idoya del Río, Tim Kelly.

Este artículo propone el uso de un metamodelo holístico como solución a los problemas que presentan los estándares a la hora de su uso en la implementación de sistemas críticos

Muchos sistemas críticos deben cumplir con las normas de seguridad como una forma de garantizar que no suponen riesgos indebidos para las personas, la propiedad o el medio ambiente.

El cumplimiento de la seguridad es una actividad muy exigente, ya que los estándares pueden constar de cientos de páginas y los profesionales generalmente tienen que demostrar el cumplimiento de miles de criterios relacionados con la seguridad.

Además, el texto de las normas puede ser ambiguo, incoherente y difícil de comprender, por lo que es difícil determinar cómo estructurar y gestionar eficazmente la información sobre el cumplimiento de la seguridad.

Estas cuestiones se vuelven aún más difíciles cuando se pretende reutilizar un sistema en otro dominio de aplicación con diferentes estándares aplicables.

Este documento tiene como objetivo resolver estos problemas proporcionando un metamodelo para la especificación de las necesidades de cumplimiento de la seguridad de los sistemas críticos.

El metamodelo es holístico y genérico, y abstrae conceptos comunes para demostrar el cumplimiento de la seguridad de diferentes estándares y dominios de aplicación. Su aplicación resulta en la especificación de "marcos de referencia de aseguramiento" para sistemas críticos de seguridad, que corresponden a un modelo de los criterios de seguridad de una norma dada.

Para validar el metamodelo con estándares de seguridad, partes de varias normas han sido modeladas por personal académico y de la industria, y se han analizado otras normas. También se ha recibido retroalimentación de profesionales, incluyendo retroalimentación durante un taller.

Los resultados de la validación muestran que el metamodelo puede usarse para especificar las necesidades de cumplimiento de seguridad para

aeroespacial, automoción, aviónica, defensa, sanidad, maquinaria, marítimo, petróleo y gas, industria de procesos, ferrocarril y robótica.

Los profesionales consideran que el metamodelo puede satisfacer sus necesidades y obtener beneficios en su uso.

Como conclusión del artículo el metamodelo apoya la especificación de las necesidades de cumplimiento de seguridad para la mayoría de los sistemas críticos basados en computadoras y software intensivo. Los modelos resultantes pueden proporcionar un medio eficaz de estructurar y administrar la información de cumplimiento de seguridad.

5. ***S-TunExSPEM: Hacia una extensión de SPEM 2.0 a los procesos orientados a la seguridad ajustables de modelo y de intercambio.*** Autores: Barbara Gallina, Karthik Raja Pitchai, and Kristina Lundqvist.

Este trabajo propone S-TunExSPEM, una extensión del Metamodelo 2.0 de SPEM 2.0 para permitir a los usuarios especificar procesos orientados a la seguridad para el desarrollo de sistemas críticos de seguridad en el contexto de la seguridad.

Normas de seguridad basadas en procesos (por ejemplo, EN 50128, DO-178B, Etc.) incorporan las mejores prácticas a ser adoptadas para desarrollar sistemas o software críticos para la seguridad.

En algunos dominios, el cumplimiento de las normas es necesario para obtener el certificado de las autoridades de certificación. Por lo tanto, una interpretación bien definida de los procesos a ser adoptados es esencial para la certificación.

En la actualidad, ningún medio satisfactorio permite a los ingenieros de procesos y a los administradores de seguridad modelar y cambiar los procesos orientados a la seguridad.

Para superar esta limitación, este trabajo propone S-TunExSPEM, una extensión del Metamodelo SPEM 2.0 para permitir a los usuarios especificar procesos orientados a la seguridad para el desarrollo de sistemas críticos de acuerdo con el nivel de seguridad requerido.

Además, para habilitar el intercambio para simulación, monitoreo y ejecución, los conceptos de S-TunExSPEM se asignan a los conceptos de XML Process Definition Language 2.2 (XPDL 2.2).

6. ***Ontology-Based Identification of Commonalities and Variabilities Among Safety Processes.*** Autores: Barbara Gallina y Zoltan Szatmari.

Este artículo propone resolver el problema que tienen los fabricantes a la hora de interpretar y cumplir adecuadamente los estándares de seguridad de los sistemas de seguridad críticos, creando modelos de estándares de seguridad basados en ontologías y automatizando el trabajo comparativo.

Las normas de seguridad imponen requisitos en el proceso de desarrollo de los sistemas críticos de seguridad. Para fines de certificación, los fabricantes tienen que interpretar y cumplir adecuadamente estos requisitos, que presentan variabilidades.

De manera más específica, se pueden identificar los puntos comunes y las variabilidades al comparar diferentes niveles de criticidad dentro de la misma versión de un solo estándar, versiones diferentes de los mismos estándares o diferentes estándares dentro del mismo dominio o incluso dentro de diferentes dominios.

El tiempo y el costo requeridos para realizar el trabajo comparativo aumentan al pasar de una única versión a diferentes estándares dentro de diferentes dominios. Esto se debe al uso de diferentes términos, que a veces no indican una semántica diferente.

Estas diferencias ralentizan no sólo el suministro de entregables, sino también la auditoría de dichos productos. Identificar los puntos en común y las variabilidades es crucial para permitir a los fabricantes acelerar la creación de suministros relacionados con el proceso a través de la reutilización sistemática.

Al mismo tiempo la reutilización bien definida y administrada, acelerar el proceso de auditoría en el lado de la autoridad de certificación.

Los autores afirman que los puntos en común entre la seguridad y la protección son frecuentemente oscurecidos por el uso de diferentes conceptos y terminologías.

De hecho, existe una variación considerable en la terminología dentro y entre las comunidades de seguridad y protección. Por lo tanto, para lograr un entendimiento común de los conceptos clave dentro de cada dominio, hay una necesidad de establecer una lengua franca o incluso una ontología común.

En este trabajo, para facilitar la identificación y sistematización de los aspectos comunes y las variabilidades, se propone un nuevo método denominado OPER, que es el de la Reutilización de los Elementos de Procesos Basados en Ontologías.

En este método, se propone proporcionar modelos basados en la ontología (dados en conformidad con OWL2.0) relacionados con los procesos de seguridad obligatorios dentro de las normas, a continuación, semi-automatizar la identificación de los comunes y variabilidades.

OPER, un método novedoso que permite a los usuarios: (1) referirse a una lengua franca común al proceso (2) semi-automatizar la comparación de estándares, y (3) generar modelos SoPL a partir de modelos de procesos de seguridad representados a través de ontologías.

7. ***Safety Oriented Software Engineering Process for Autonomous Robots.***
Autores: Vladislav Gribov y Holger Voos.

En este trabajo se propone un proceso de ingeniería de software basado en modelos orientados a la seguridad para robots autónomos.

La atención se centra en el modelado de la caja de seguridad basado en la norma ISO / DIS 13482. En combinación con una arquitectura de software del robot de múltiples capas de seguridad que permite rastrear los requisitos de seguridad y para modelar las propiedades relevantes de seguridad en las etapas tempranas del diseño con el fin de construir una cadena de evidencia confiable.

La segregación física de los robots y los seres humanos funciona bien en un entorno industrial con manipuladores de robots estacionarios, acompañado de los correspondientes entrenamientos de seguridad para el personal.

Por razones obvias, tales medidas no son posibles si se requiere pHRI (interacción físico humano-robot) por la aplicación del robot y si no es posible un diseño de robot intrínsecamente seguro.

Los robots autónomos y los seres humanos (no entrenados) que comparten el espacio de operación y cooperan entre sí, crean nuevos tipos de riesgos y requisitos con respecto a la seguridad de los robots.

La interacción humano-robot segura durante la operación autónoma se hace esencial y necesaria, pero también requiere nuevas normas de seguridad.

La solicitud de una nueva norma de seguridad es seguida por la solicitud de un proceso de ingeniería práctico correspondiente, especialmente para el software del robot (SW). Los robots de servicio existentes, tanto comerciales como las soluciones académicas, se diseñan sobre todo de una manera más intuitiva sin proporcionar un proceso de ingeniería claro y seguridad-orientado para el diseño del robot.

Los planes futuros para un amplio despliegue de robots de servicio y de cuidado personal también sugieren costos limitados para la fase de desarrollo del robot.

Sin embargo, esto sólo es posible si se aplica un proceso de ingeniería que se ocupa de los requisitos de seguridad de una manera formal y reutiliza las soluciones de seguridad.

8. ***Towards a Common Safety Ontology for automobiles and Railway vehicles***. Autores: Bernhard Hulin, Hermann Kaindl, Thomas Rathfux, Roman Popp, Edin Arnautovic, Roland Becker.

Este artículo trata sobre la investigación y búsqueda de una ontología común para automóviles y vehículos ferroviarios, en cuanto a normas específicas de seguridad.

Los automóviles y vehículos ferroviarios tienen cada uno sus normas de seguridad específicas respectivamente. Sin embargo, ambos son vehículos terrestres y por lo tanto comparten un gran conjunto de peligros comunes y tipos de accidentes. Esto requiere una ontología de seguridad común que cubra ambos dominios.

Aparte de muchos puntos comunes entre los automóviles y vehículos ferroviarios, encontramos algunas diferencias importantes entre las normas de seguridad de estos dominios, en particular entre las normas ISO 26262, EN 50126 y SIRF (la Norma alemana para vehículos ferroviarios).

Basándose en sus respectivos glosarios, los autores intentan resolver ciertas diferencias. Esto les llevó a un conjunto común de conceptos formalizados y sus relaciones.

Partiendo del análisis de la terminología (en parte conflictiva) de las normas de seguridad relacionadas, se demuestra el desarrollo de modelos conceptuales y su fusión con un modelo ontológico común. Este enfoque puede servir como base para una ontología de seguridad común, por ejemplo, para sistemas automotrices y ferroviarios.

El trabajo futuro tendrá que extender y evolucionar los modelos ontológicos y para incluir una ontología superior apropiada. Los modelos extendidos proporcionarán apoyo para al menos análisis / clasificación de peligros. También los estudios de casos en las áreas de aplicación potenciales serán importantes para la validación de nuestro enfoque propuesto.

Los autores consideran esto como un paso importante hacia una ontología común para automóviles y vehículos ferroviarios. Dicha ontología debería facilitar la reutilización de los análisis de riesgos y riesgos de un dominio a otro y debería tener importantes áreas de aplicación.

9. ***A UML Profile for the Development of IEC 61508 Compliant Embedded Software.*** Autores: Dirk Kuschnerusy, Felix Brunsy, Attila Bilgic and Thomas Muschy.

Este trabajo propone un perfil UML que se extiende el Unified Modeling Language (UML) para apoyar el desarrollo de software embebido crítico para la seguridad de conformidad con la norma de seguridad IEC 61508.

Esta mejora de densidad de información en los modelos de software puede explotarse como base para actividades en diversas fases de desarrollo de software, por ejemplo, la reutilización de componentes de software certificados o el despliegue de componentes de software críticos y no críticos a nodos separados.

El software integrado controla y supervisa procesos y aplicaciones en diversos dominios, por ejemplo, industria de procesos, automoción y transporte.

En muchos de los procesos controlados el fallo del sistema bajo control o sólo partes del sistema puede conducir a serias amenazas tanto para el medio ambiente como para los seres humanos.

En consecuencia, estos dominios tienen una alta demanda de medios que reduzcan los riesgos existentes a un nivel tolerable. Estas funciones de seguridad garantizar la seguridad funcional del proceso y, a menudo son implementados por software integrado de seguridad crítico.

Para orientar a los desarrolladores, las autoridades de certificación y los clientes en el ámbito de la seguridad funcional, existen diferentes normas generales y específicas del dominio

El aumento de la aplicación de software embebido para las tareas relacionadas con la seguridad y la demanda de los clientes para una cartera más amplia de dispositivos de medida con respecto a ambas características y aplicaciones conducen a un aumento de la complejidad del software.

Además, la industria se enfrenta a la necesidad de desarrollar software embebido de alta calidad en una cantidad minimizada de tiempo.

Una solución natural a esta situación es mejorar los procesos de software hacia una mayor eficiencia en la reutilización de software. Los enfoques populares como el desarrollo basado en componentes y las líneas de productos de software lo hacen derivando una plataforma de software que integra módulos de software intercambiables.

El UML es ampliamente aceptado por la industria y la investigación como estándar de facto en el modelado de software orientado a objetos. También es una tecnología clave en el campo de la investigación activa de desarrollo impulsado por el modelo, que se centra en modelos como elementos centrales en el desarrollo de software. Soporta numerosos enfoques y ofrece la posibilidad de modelar componentes de software y su despliegue en nodos y se menciona como método semi-formal recomendado para diseño de arquitectura de software y diseño detallado.

El artículo, ofrece una visión general de un perfil UML para el desarrollo de sistemas embebidos críticos para la seguridad conforme a la norma IEC 61508.

El perfil tiene como objetivo aumentar la productividad del desarrollo de software basado en modelos con respecto a la consecución del cumplimiento estándar, reutilización de artefactos de software y documentación de la información de certificación.

10. ***Towards a Safer Development of Driver Assistance Systems by Applying Requirements-Based Methods.*** Autores: Henning Jost, Silke Kohler and Frank Koster.

    La finalidad de este artículo reside en la presentación de una serie de métodos, basados en las actividades de ingeniería de requisitos, que tienen como objetivo apoyar al desarrollo y a la calificación de los sistemas críticos de seguridad con el fin de lograr un desarrollo más seguro, dentro del campo de los sistemas de transporte inteligentes.

    El desarrollo de sistemas de transporte inteligentes, como los sistemas de asistencia al conductor se enfrenta actualmente el aumento de la complejidad del sistema y un número cada vez mayor de requisitos, por ejemplo, de las normas de seguridad.

    Por lo tanto, las actividades de ingeniería de requisitos ganan un papel cada vez más importante cuando se trata del desarrollo de sistemas de transporte críticos para la seguridad.

    A modo de ejemplo, la introducción de la norma ISO 26262 en el sector de la automoción requiere la adaptación de los procesos de desarrollo industrial existentes para cumplir con la próxima norma.

    Los métodos propuestos abordan la nueva demanda en la gestión de requisitos para un desarrollo más seguro de los sistemas de asistencia al conductor.

    Mediante ontologías, se formalizan requisitos de dominio como ISO 26262 que proporcionan un modelo de referencia para soportar el descubrimiento semi-automatizado de requisitos.

Los métodos propuestos han sido implementados en un prototipo de cadena de herramientas. En cuanto a los sistemas de asistencia al conductor, se utiliza un sistema de alerta de salida de carril como ejemplo de aplicación para ilustrar el procedimiento.

Los métodos presentados muestran el potencial de los artefactos de ingeniería de requisitos, como la trazabilidad, para propósitos diferentes a los originalmente previstos, abordando los problemas identificados al principio del documento.

Se demuestra también que la identificación de los requisitos estándar y la adaptación a un conjunto aplicable de requisitos de producto y proceso pueden ser parcialmente automatizados y, por lo tanto, facilitarse utilizando una representación formalizada de estándares, como la ISO 26262.

11. ***Extracting Models from ISO 26262 for Reusable Safety Assurance.*** Autores: Yaping Luo, Mark van den Brand, Luc Engelen, John Favaro, Martijn Klabbers, y Giovanni Sartori.

En este artículo se propone un enfoque basado en modelos para asegurar el cumplimiento de las normas de seguridad para facilitar la reutilización en los procesos de evaluación, calificación y certificación.

A medida que el software es cada vez más complejo, se despliega en sistemas críticos la seguridad operacional, el desafío de evaluar la seguridad de los sistemas de acuerdo con las normas pertinentes está creciendo.

Debido al extenso trabajo manual requerido, la validación del cumplimiento de estos sistemas con las normas de seguridad es una actividad costosa y que requiere mucho tiempo. Además, a medida que los productos evolucionan, puede ser necesaria una nueva evaluación.

Por lo tanto, la obtención de datos de aseguramiento reutilizables para la evaluación o reevaluación de la seguridad es muy deseable.

Se describen tres técnicas de modelado diferentes. Se introduce un modelo de estructura para describir la estructura general de la norma. Se utiliza una técnica basada en reglas para extraer el modelo conceptual. Y una estructura del software y el metamodelo de ingeniería de procesos de sistemas proporciona una descripción de sus procesos.

Finalmente, la validación en el contexto de un caso de uso concreto en el proyecto FP7 OPENCOSS muestra que los modelos resultantes de la

aproximación se parecen a los modelos industriales, pero que, inevitablemente, requieren el ajuste fino de los expertos en el dominio.

En este trabajo, se presenta un enfoque basado en el modelo de seguridad que permite la reutilización de aseguramiento a través del modelado objetivo y el costo-e de las normas pertinentes.

La metodología de la bola de nieve proporciona reglas para extraer el modelo conceptual de un estándar de seguridad, reduciendo así la cantidad de trabajo manual.

Más del 90% de los conceptos en los modelos industriales están cubiertos por el modelo conceptual. Se obtendrá un mejor resultado si los expertos de dominio están involucrados en todas las etapas del enfoque de la bola de nieve, pero será más costoso.

Además, la disponibilidad de un modelo generado reduce por tanto como 80% semi-automáticamente la cantidad de tiempo para validar el modelo final por los expertos en el dominio.

El proceso en el estándar se modela con la SPEM de OMG. Aunque el enfoque que actualmente opera sólo en un nivel muy alto, que proporciona una base para la descripción de un modelo de proceso en el contexto de las normas de seguridad.

12. ***Ensuring Safety of Avionics Software at the Architecture Design Level An Industrial Case Study.*** Autores: Ji Wu, Tao Yue, Shaukat Ali, Huihui Zhang.

Este artículo presenta una metodología de modelado que incluye un perfil UML para especificar requisitos de seguridad en un modelo de arquitectura basado en componentes y un conjunto de directrices de diseño en software de aviónica.

Asegurar que el software de aviónica cumple con los requisitos de seguridad en cada etapa de desarrollo es muy importante para garantizar el funcionamiento seguro de un sistema de aviónica.

Muchos requisitos de seguridad son impuestos por varias normas y regulaciones industriales que deben ser satisfechas por el software de aviónica.

Una de estas normas es la DO-178B / C, que proporciona directrices (por ejemplo, el proceso de desarrollo y los objetivos a satisfacer en las actividades de desarrollo) para satisfacer los requisitos de seguridad.

Este artículo presenta una metodología de modelado que incluye un perfil UML para especificar requisitos de seguridad en un modelo de arquitectura basado en componentes y un conjunto de directrices de diseño en software de aviónica.

Estos requisitos de seguridad fueron identificados de ambas normas (principalmente DO-178B / C) y las prácticas de ingeniería actuales en el dominio del sistema de aviónica.

La metodología hace cumplir automáticamente los requisitos de seguridad. Hemos aplicado la metodología en un sistema de piloto automático industrial y varias fallas previamente no detectadas fueron reveladas.

En el presente trabajo, se ha propuesto la arquitectura orientada a la seguridad de modelado (SOAM) para el modelado de la arquitectura de software de aviónica desde el aspecto de la seguridad.

Los requisitos de la arquitectura SOAM se identifican por análisis de dominio sistemática y revisar estándar DO-178B / C.

Los objetivos de SOAM son la captura de los requisitos de seguridad que se requieren en DO-178B / C, proporcionando “SafetyProfile” y un conjunto de directrices para apoyar el modelado de la arquitectura para satisfacer los requisitos de seguridad.

Para evaluar nuestro enfoque, se llevó a cabo un estudio de caso sobre el sistema de piloto automático industrial. Todos los estereotipos propuestos en SafetyProfile se aplican al menos una vez.

Mediante la aplicación de la SOAM, la arquitectura funcional del sistema de piloto automático fue refinado y todas las 32 propiedades de seguridad identificados se verificaron mediante la evaluación del modelo de arquitectura en contra de las restricciones OCL.

Seis fallos implícitos previamente no detectadas fueron identificados durante el proceso de modelado de arquitectura. Estas propiedades de seguridad y los estereotipos capturados en SOAM no son de ninguna manera específicos para el caso de estudio presentado en este artículo. Se pueden aplicar a cualquier software de aviónica cuyo desarrollo se requiere para ajustarse a la norma DO-178B / C.

13. ***Toward DO-178B-compliant Test Models.*** Autores: Heiko Stallbaum, Mark Rzepka.

En este artículo, se aborda un inconveniente importante para la aplicación de MBT (Model-based Testing) en el dominio aviónica: el cumplimiento DO-178B.

Model-based Testing (MBT) ayuda a manejar la creciente complejidad del software y reduce el esfuerzo de desarrollo mediante el soporte de herramientas y la automatización.

En este artículo, se aborda un inconveniente importante para la aplicación de MBT en el dominio aviónica: el cumplimiento de DO-178B. DO-178B es el estándar más relevante para el desarrollo de software de aviónica en aviones civiles.

Tiene un fuerte énfasis en las pruebas basadas en requisitos y la trazabilidad de los datos del ciclo de vida, pero no considera nuevas metodologías de desarrollo como MBT.

Para facilitar la aplicación de MBT en el campo de la aviónica, se introduce en este artículo un nuevo perfil UML que puede utilizarse para extender modelos de prueba UML estándar para MBT con información relevante para la certificación DO-178B.

Por lo tanto, estos modelos de prueba pueden servir como artefactos de apoyo en la certificación DO-178B y proporcionan un soporte explícito para la certificación DO-178B en caso de que se aplique MBT.

Algunos de los beneficios del perfil UML para los modelos de prueba compatibles con DO-178B se ilustran en un escenario de aplicación.

En la solución propuesta por el artículo, primero se ha analizado cuidadosamente el estándar DO-178B para identificar los requisitos esenciales para los modelos de prueba compatibles con DO-178B.

Resumiendo, los resultados de este análisis, se puede subrayar que DO-178B tiene un fuerte énfasis en las pruebas basadas en requisitos y la trazabilidad de los datos del ciclo de vida. Sobre la base de los resultados del análisis hemos definido un nuevo perfil UML que proporciona soporte de certificación explícita DO-178B.

Mediante este perfil UML, los modelos de prueba estándar pueden ampliarse con información relevante para la certificación y servir como artefactos de soporte en la certificación DO-178B.

Como ejemplo, se implementa el nuevo perfil UML para diagramas de actividad UML.

Finalmente, se ha esbozado la aplicación del perfil UML propuesto en un modelo de prueba para un sistema de piloto automático y también hemos demostrado cómo el soporte de certificación DO-178B puede ser dado por el perfil.

14. ***Employing early model-based safety evaluation to iteratively derive EE architecture design.*** Autores: Rupano, C. Buckl, L. Fiege, M. Armbruster, A. Knoll, G. Spiegelberg.

En este artículo se presenta y se discute un proceso de diseño y refinamiento de arquitectura iterativa que se centra en los requerimientos del estándar ISO 26262 y en el análisis basado en modelos de métricas relacionadas con la seguridad.

El estándar ISO 26262 aborda el desarrollo de funciones seguras en los vehículos mediante la especificación de métodos potencialmente utilizados en el diseño y el ciclo de vida del desarrollo.

No indica con certeza lo que es necesario y deja espacio para la interpretación. Sin embargo, los arquitectos de los sistemas eléctricos / electrónicos necesitan límites de diseño para tomar decisiones durante el diseño evolutivo de la arquitectura sin añadir un riesgo de cambios tardíos.

La selección correcta de los mecanismos de seguridad de las alternativas en las primeras fases de diseño es vital para el tiempo de comercialización de los sistemas críticos.

En este artículo se presenta y se discute un proceso de diseño y refinamiento de arquitectura iterativa que se centra en los requerimientos del estándar ISO 26262 y en el análisis basado en modelos de métricas relacionadas con la seguridad.

Este proceso simplifica la identificación de las partes más sensibles de la arquitectura, la selección de los mejores mecanismos de seguridad adecuados para reducir así la tasa de fallos a nivel del sistema y mejorar las métricas definidas por la norma.

Para apoyar el proceso definido presentamos los metamodelos que se pueden integrar con los marcos existentes de DSL (lenguaje específico del dominio) para ampliarlos con información que apoya la extracción adicional del comportamiento de propagación de fallos.

Se proporciona un marco para el análisis de modelos de arquitectura y selección de mecanismos de seguridad. También se proporcionan detalles sobre el conjunto de herramientas basado en modelos que se ha desarrollado para soportar los métodos de análisis y síntesis propuestos, con la finalidad de demostrar su aplicación al análisis de un modelo de sistema de dirección por cable y la selección de mecanismos de seguridad para ello.

En este trabajo se ha presentado un método que evalúa las opciones de diseño temprano en proceso de desarrollo y de forma iterativa añadir detalles arquitectura hasta una arquitectura segura y rentable de la especie se deriva.

Se proponen repositorios de modelos para recoger una parte importante del conocimiento del dominio (cobertura de los mecanismos de seguridad y compensaciones de recursos asociado), por lo que el diseño de decisiones puede llegar a ser más determinista.

Al mismo tiempo, las instancias de modelos de repositorios se pueden utilizar para llevar a cabo evaluación de la seguridad después de un flujo de trabajo bien definido. Como efecto, la evidencia y argumentos sobre los atributos esenciales de seguridad, como los modelos de cobertura y de fallo, se acumulan de forma continua dentro de la infraestructura de diseño, lo que hace que la reutilización de estos datos, e incluso de toda la arquitectura de referencia, es posible.

Para proporcionar el máximo beneficio, el enfoque presentado se tiene que aplicar en un proceso de desarrollo basado en modelos.

En este entorno se proporciona una alta trazabilidad de los requisitos de entrada a la aplicación y la argumentación cuantitativa necesaria para la certificación ISO 26262.

La evaluación preliminar con un prototipo de herramienta muestra que la evolución iterativa permite la concentración en un alto nivel de modelado (por lo tanto, reduciendo el nivel requerido de conocimientos) para tomar decisiones de diseño.

15. ***Supporting the verification of compliance to safety standards via model-driven engineering.*** Autores: Rajwinder Kaur Panesar-Walawege, Mehrdad Sabetzadeh, Lionel Briand.

En este trabajo se propone un nuevo enfoque para ayudar a los proveedores en la creación de las pruebas necesarias para la certificación de acuerdo con las normas.

Muchos sistemas críticos para la seguridad están sujetos a la certificación de seguridad como una forma de asegurar que estos sistemas no pueden dañar indebidamente a la gente, la propiedad o el medio ambiente.

La creación de la evidencia necesaria para la certificación puede ser una tarea difícil debido al tamaño de los estándares textuales basados en la certificación que se realiza y la facilidad de estos estándares a la interpretación subjetiva.

En este trabajo se propone un nuevo enfoque para ayudar a los proveedores en la creación de las pruebas necesarias para la certificación de acuerdo con las normas.

El enfoque se basa en la ingeniería dirigía por modelos (MDE) y aborda los desafíos del uso de estándares de certificación al tiempo que proporciona asistencia con el cumplimiento.

Dada una norma de seguridad, se construye un modelo conceptualque proporciona una interpretación concisa y explícita de la norma. Este modelo se utiliza para crear un perfil UML que ayuda a los proveedores del sistema a relacionar los conceptos del estándar de seguridad con los del dominio de aplicación, permitiendo a los proveedores demostrar cómo sus artefactos de desarrollo del sistema cumplen con el estándar.

Se proporcionan una solución generalizable y una herramienta para apoyar la verificación del cumplimiento de las normas de seguridad.

La validación empírica del trabajo se presenta a través de un estudio de caso industrial que muestra cómo los conceptos de un sistema de control de producción submarino pueden alinearse con los requisitos de evidencia de la norma IEC61508. Un estudio posterior examina las percepciones de los profesionales sobre la solución.

El estudio de caso indica que la empresa proveedora, donde se realizó el estudio encontró que el enfoque útil para ayudar a prepararse para la certificación de su software.

La encuesta indica que los profesionales encontraron que el enfoque propuesto era fácil de entender y que estarían dispuestos a adoptarlo en la práctica. Dado que la norma IEC61508 se aplica a múltiples dominios, estos resultados sugieren una mayor aplicabilidad y utilidad de nuestro trabajo.

3.- APROXIMACIÓN PARA LA REPRESENTACIÓN DE ESTÁNDARES DE SEGURIDAD:

En este apartado se explica cómo realizar la aproximación para la representación de estándares de seguridad que se ha definido a partir del metamodelo propuesto en (José Luis de la Vara, Alejandra Ruiz, Katrina Attwood, Huáscar Espinoza, Rajwinder Kaur Panesar-Walawege, Ángel López, Idoya del Río, Tim Kelly, 2016) y la herramienta KM.

3.1.- Propuesta para Representar las Normas de Seguridad con KM:

La propuesta de representar estándares de seguridad con tecnologías semánticas se basa en dos elementos principales: KM, como enfoque de apoyo y herramienta para la especificación semántica de la información de una norma, y un metamodelo genérico holístico para la especificación de las necesidades de cumplimiento de seguridad.

El metamodelo indica los tipos de elementos que deben tenerse en cuenta al tener que demostrar el cumplimiento de los estándares de seguridad, así como las relaciones entre ellos.

El propósito general de la propuesta consiste en proporcionar orientación sobre cómo la terminología de una norma, los elementos de datos de los tipos de elementos y las relaciones entre los elementos se pueden representar con KM.

3.2.- Presentación del metamodelo:

Esta sección presenta el metamodelo que proponemos como referencia para la especificación de las necesidades de cumplimiento de la seguridad en forma de RAF (Reference Assurance Framework).

El metamodelo incluye los conceptos claves y las relaciones entre ellos para demostrar cumplimiento de seguridad. Captura las nociones abstractas que pueden usarse para describir la información que necesita recogerse para mostrar el cumplimiento de los estándares de seguridad y para manejar el cambio de un sistema. (José Luis de la Vara, Alejandra Ruiz, Katrina Attwood, Huáscar Espinoza, Rajwinder Kaur Panesar-Walawege, Ángel López, Idoya del Río, Tim Kelly, 2016).

Específicamente, el metamodelo RAF corresponde a un medio unificado para la creación de modelos de aseguramiento y certificación de la seguridad. *(Ibídem).*

La razón principal para desarrollar modelos RAF es crear una interpretación consistente de la norma que se está utilizando y vincular la interpretación al producto que se está certificando. *(Ibídem).*

La necesidad de una interpretación consistente deriva del hecho de que las normas de seguridad son documentos textuales susceptibles de interpretación subjetiva. Al crear un modelo, no evitamos la subjetividad, pero ayudamos a desarrollar y comunicar una interpretación compartida y consistente. *(Ibídem).*

El metamodelo RAF también proporciona un medio común para comparar los estándares de seguridad, basados en una terminología común (es decir, las clases y asociaciones del metamodelo), y puede a su vez facilitar la reutilización de la seguridad de seguridad a través de estándares de seguridad y dominios de aplicación. *(Ibídem).*

La Figura. 1 muestra el metamodelo RAF. Para mayor claridad, se ha descompuesto el metamodelo en cuatro partes interrelacionadas: la jerarquía de especialización, los componentes principales de un RAF, las asociaciones de elementos de referencia y la información sobre la aplicabilidad de la RAF. (No se muestran los atributos de las clases en todas las partes para mantener la figura lo más pequeña y sencilla posible). *(Ibídem).*

Un elemento de referencia (figura 2 (a)) tiene un ID, un nombre, una descripción y una referencia. La referencia es para especificar desde donde se modela un elemento (por ejemplo, una cláusula de un estándar de seguridad). *(Ibídem).*

Algunos elementos de referencia son también elementos de aseguramiento de referencia, que representan los principales aspectos de seguridad que deben modelarse para el ciclo de vida de un sistema crítico:

- Requisito de referencia: condiciones (por ejemplo, un objetivo) que podrían tener que cumplirse (por ejemplo, el software se producirá para lograr modularidad, estabilidad y capacidad de modificación segura, en IEC 61508).

- Actividad de referencia: unidad de comportamiento que podría tener que ser ejecutada (por ejemplo, procesos de desarrollo de software en DO-178C).

- Rol de referencia: tipo de agentes que podrían tener que estar involucrados (por ejemplo, Diseñador en EN-50128).

- Artefacto de referencia: unidad de datos que podría tener que ser administrada (por ejemplo, plan de seguridad en ISO 26262).

- Técnica de referencia: forma específica de ejecutar una actividad de referencia o crear un artefacto de referencia (por ejemplo, métodos formales en IEC 61508).

• Relación de artefacto de referencia: relación entre dos artefactos de referencia que podría tener que ser grabada (por ejemplo, satisface la descripción del diseño DO-178C satisface los datos de requisitos de software).

• Atributo de referencia de artefacto: característica de un artefacto de referencia que podría tener que ser registrada (por ejemplo, el resultado esperado de un caso de prueba en la norma EN 50128).

El marco de referencia de aseguramiento corresponde a una composición de los criterios de seguridad con los que el ciclo de vida de un sistema crítico podría tener que mostrar cumplimiento (Fig. 2 (b)).

Intuitivamente, un RAF representa un estándar y más concretamente sus necesidades de cumplimiento de seguridad. Además de los requisitos de referencia, las actividades, las técnicas y los artefactos, el RAF puede consistir en un tipo de criticidad de referencia (categoría de criterios de reducción de riesgos) y un tipo de aplicabilidad de referencia (categoría de relevancia o idoneidad para los elementos de referencia). *(Ibídem).*

Por ejemplo, el SIL (nivel de integridad de seguridad) es probablemente el tipo de criticidad de referencia más conocido. Se utiliza en IEC 61508 y otras normas relacionadas (por ejemplo, EN 50128). Estas normas también suelen proporcionar alguna forma de recomendación de uso (tipo de aplicabilidad de referencia) para las técnicas de referencia. *(Ibídem).*

Los elementos de referencia asociables están relacionados entre sí de varias maneras (figura 2 (c)). Los requisitos de referencia, que se pueden descomponer en sub-requisitos, se pueden asignar a algún elemento de referencia restringido. En otras palabras, las actividades de referencia, artefactos, roles y técnicas pueden ser responsables del cumplimiento de los requisitos de referencia. *(Ibídem).*

Los roles de referencia pueden participar en actividades de referencia y ser responsables de algunos artefactos de referencia. Las técnicas de referencia pueden usarse tanto para las actividades de referencia como para los artefactos de referencia, y ser especializadas en otras técnicas. *(Ibídem).*

Por ejemplo, el modelado se especializa en EN-50128 en modelado de datos y diagramas de secuencia, entre otras técnicas. *(Ibídem).*

Las actividades de referencia pueden tener artefactos de referencia de entrada y salida y relaciones de artefacto de referencia de salida. *(Ibídem).*

Una actividad de referencia puede además descomponerse en subactividades y tener actividades de referencia predecesoras y sucesoras. Los artefactos de referencia pueden tener atributos de artefacto de referencia, pueden ser el origen o el destino de las relaciones de referencia de artefactos y pueden registrar estas relaciones. Por último, se puede especificar la información sobre multiplicidad y los efectos de los cambios en el origen y destino de una relación de artefacto de referencia. *(Ibídem).*

La información sobre la aplicabilidad de un RAF (Fig. 2 (d)) representa los criterios de seguridad de las normas con respecto a las circunstancias en las que debe mostrarse el cumplimiento de los elementos de referencia y cómo hacerlo. Esta información es, sin duda, la característica más distinguible del metamodelo RAF. *(Ibídem).*

La información sobre la aplicabilidad de la RAF se proporciona en relación con algún nivel de aplicabilidad de referencia (niveles de pertinencia o idoneidad para un tipo de aplicabilidad de referencia) o algún nivel de aplicabilidad de referencia y cierto nivel de criticidad de referencia (nivel relativo de reducción de riesgo para un tipo de crítica de referencia). *(Ibídem).*

Por ejemplo, la IEC 61508 "recomienda" y "recomienda altamente" (niveles de aplicabilidad de referencia) el uso de algunas técnicas de referencia para el SIL 4 (nivel de criticidad de referencia), que está asociado con una probabilidad media de un fallo peligroso a petición de una función de seguridad (de 1e-5 a 1e-4). *(Ibídem).*

Las normas de seguridad presentan la aplicabilidad de referencia en el ámbito de las actividades de referencia, los requisitos de referencia o las técnicas de referencia (propietario de la aplicabilidad de la referencia), a menudo mediante tablas en su texto. *(Ibídem).*

La aplicabilidad de la criticidad de referencia suele corresponder a las celdas de estas tablas. Aunque la información sobre la aplicabilidad suele referirse únicamente a los elementos asegurables de referencia (elemento de la aplicabilidad de referencia, por ejemplo, filas en las tablas IEC 61508), las normas también pueden proporcionar información de aplicabilidad para varios elementos de referencia (por ejemplo, una combinación válida de técnicas en EN-50128) y para conjuntos de elementos de referencia en relación con otros (por ejemplo, un conjunto de roles de referencia que tienen que ser independientes de un rol de referencia dado). *(Ibídem).*

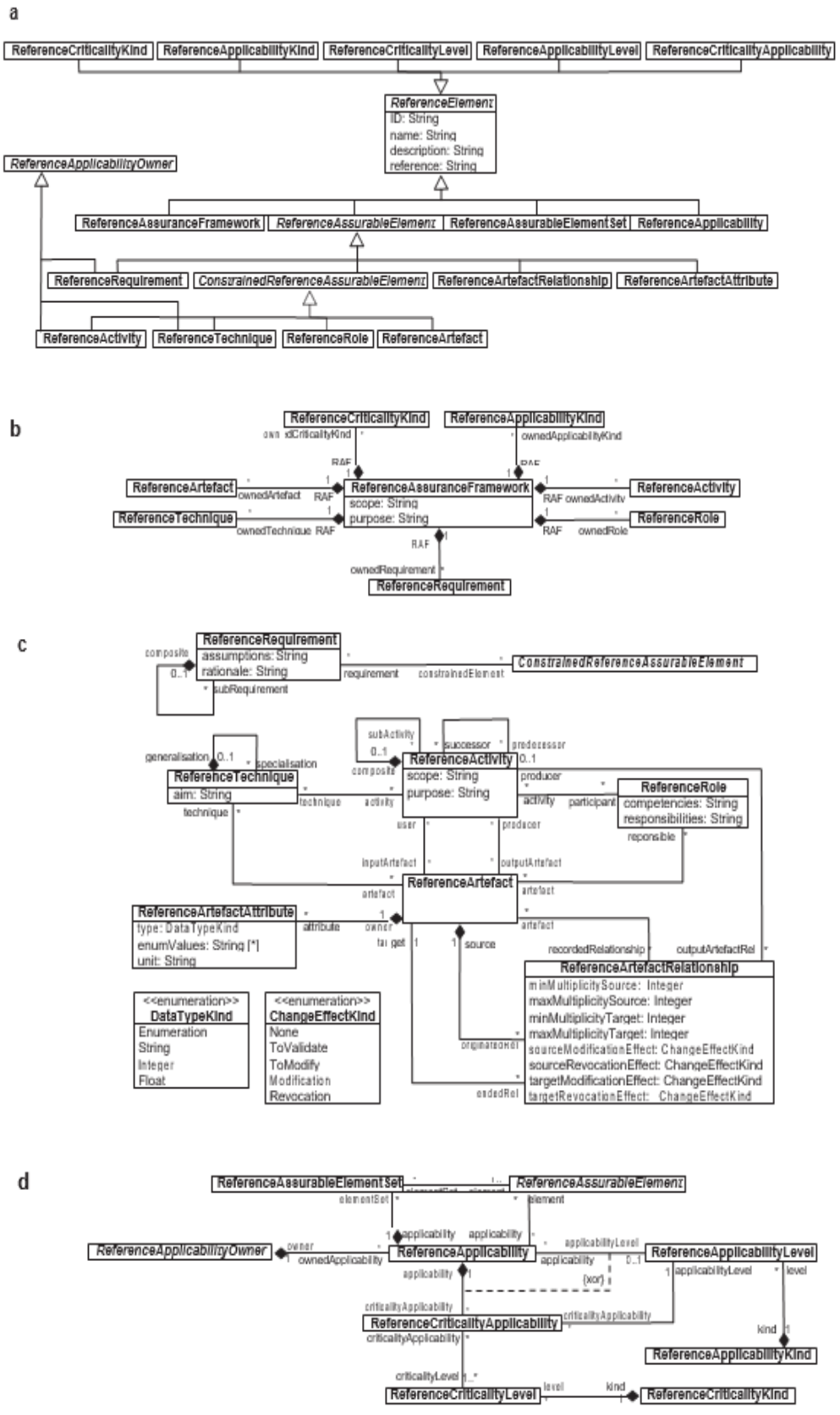


Figura 1. Metamodelo RAF.

3.3.- Capas de ontología en KM:

La figura 3. muestra la estructura de una ontología en KM. Una ontología consta de varias capas, cada una dependiendo y extendiendo la información semántica de la capa interna. (The Reuse Company, 2017).

La capa más interna (Terminología) corresponde a los términos de un dominio junto con su información sintáctica. Las relaciones entre los términos se pueden especificar en la capa del modelo conceptual, así como su semántica con los grupos; Por ejemplo, la semántica de los términos «coche» y «camión» puede ser «sistema», y especializan «vehículo». *(Ibídem).*

Los patrones se pueden desarrollar entonces para proporcionar plantillas (aka boilerplates) para la especificación de la información de un sistema; Los patrones se refieren a aspectos de las dos capas subyacentes. *(Ibídem).*

La capa de formalización incluye información acerca de cómo la información del sistema que coincide con un patrón será representada y almacenada semánticamente. *(Ibídem).*

Finalmente, en la capa de reglas de inferencia los datos de todas las otras capas pueden utilizarse para la especificación de reglas para derivar nueva información, por ejemplo, sobre la corrección de una especificación de un sistema. *(Ibídem).*

En su estado actual, la propuesta sólo se ocupa de las capas de terminología y modelo conceptual. *(Ibídem).*

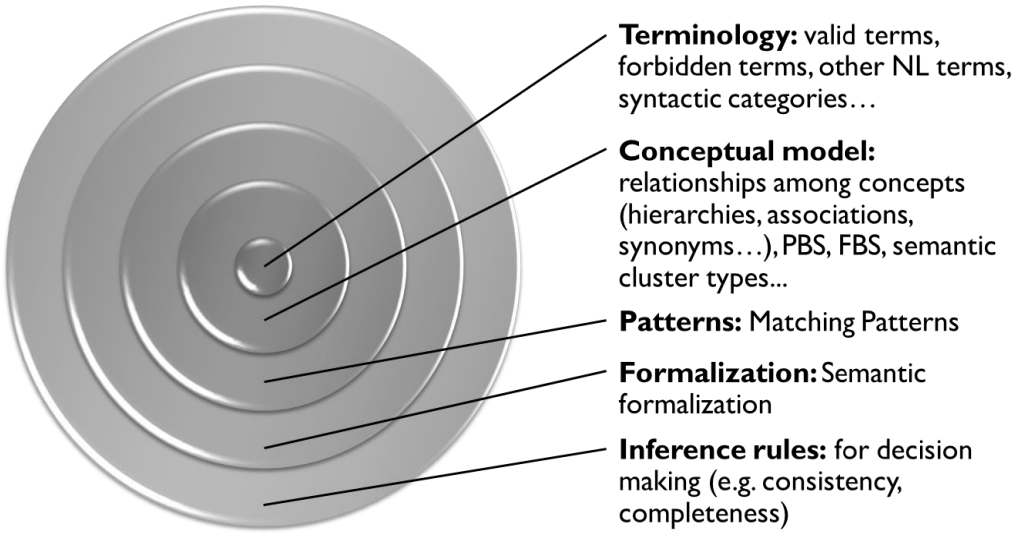


Figura 2. Capas de ontología en KM.

3.4.- Aproximación:

En esta sección se explica la metodología para aplicar la aproximación. La herramienta que se ha utilizado para llevar a cabo la aproximación ha sido Knowledge Manager (KM) descrita en el apartado 2.6.

La aproximación propuesta consta de dos actividades principales: configuración de KM y especificación de la información de una norma, que esta a su vez se divide en dos subactividades: especificación de la terminología de un estándar y especificación del modelo conceptual de una norma de seguridad.

Cada actividad consta de varios pasos, como se explica a continuación.

1ª Fase: Configuración KM:

Esta actividad es necesaria para adaptar el uso predeterminado de KM para representar estándares de seguridad, es decir, ciertos aspectos de KM deben configurarse para que un usuario pueda crear una representación adecuada de acuerdo con el metamodelo genérico holístico.

La configuración se centra en los aspectos semánticos de las normas que deben incluirse en la representación. Estos aspectos son específicos de las normas de seguridad, pero son independientes del estándar específico a representar. Deben realizarse dos tareas.

1.1 Especificación de grupos semánticos:

Los nuevos clústeres deben agregarse a la capa del modelo conceptual para poder indicar el tipo de información que representa un término.

En primer lugar, es necesario un clúster con el nombre del estándar de seguridad que se va a representar para especificar más adelante que un término cae dentro del alcance de la norma.

En segundo lugar, se introducirán los tipos de clúster pertenecientes al metamodelo. Estos clústeres tendrán la finalidad de modelar los diferentes elementos de una RAF.

Su representación en el metamodelo viene dada por rectángulos. Los tipos de clústeres son los siguientes:

- Reference Artefact.
- Reference Activity.
- Reference Role.
- Reference Artefact Atribute.
- Reference Technique.

-*Comandos para introducir clústeres en KM*:

1.- Click derecho dentro de la tabla de términos en la sección de terminología.
2.- A continuación, se desplegarán una serie de opciones. Se dará click a la opción "añadir nuevo término".
3.- Click derecho en el espacio en blanco perteneciente a la sección Clúster(s) y click en "añadir nuevo clúster".
4.- Click derecho dentro de la tabla "clústeres" y click en "añadir nuevo clúster".
5.- Introducir nombre del clúster y click en aceptar. De esta forma se irán introduciendo uno a uno los clústeres enunciados.

-*Nota*: El tipo de clúster Reference Requierement no se representará mediante esta herramienta debido a que no tiene alcance a la extensión de sus términos. Este tipo de clúster será tratado posteriormente con otra herramienta que citaremos en el siguiente punto.

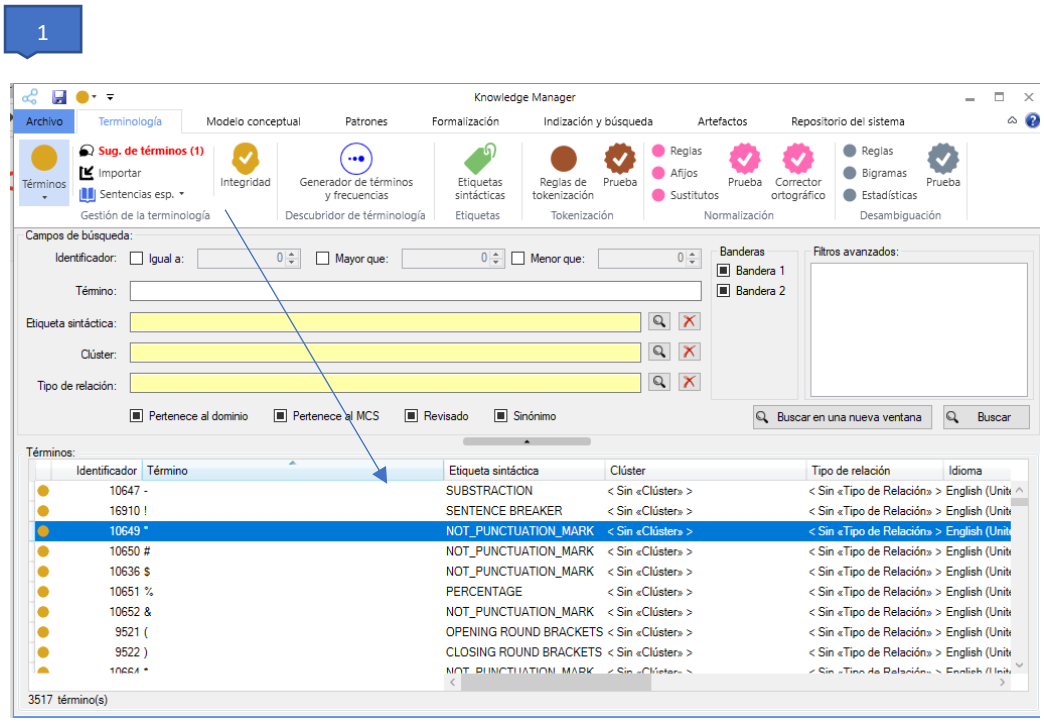


Figura 3. Paso 1: Especificación de grupos semánticos.

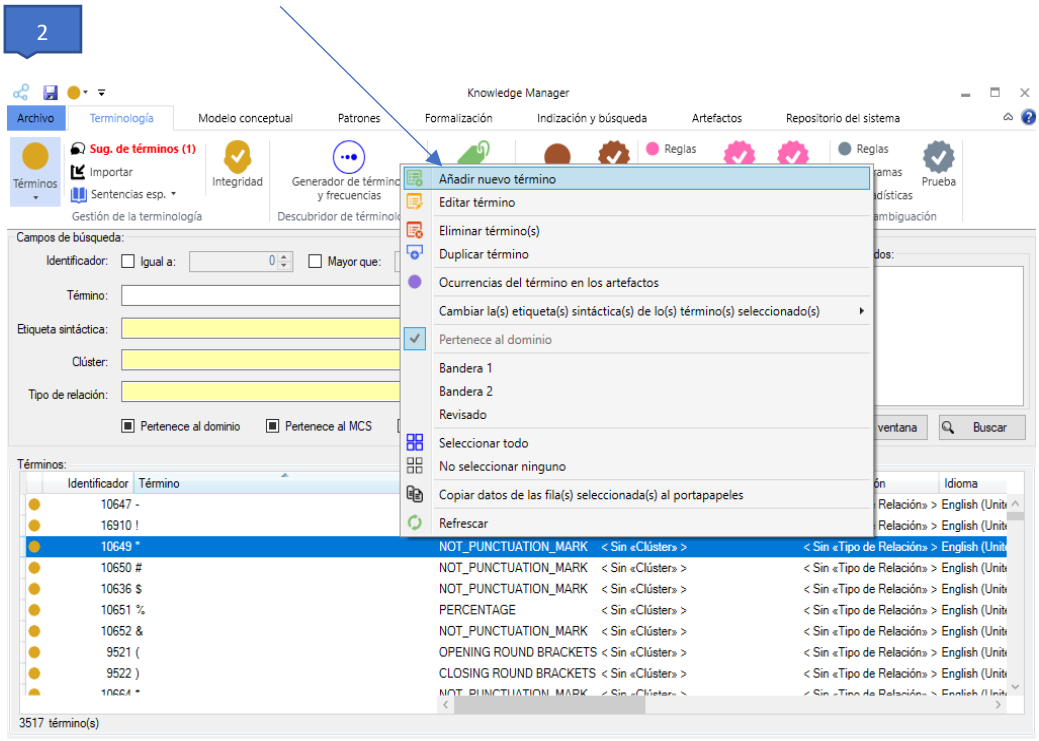


Figura 4. Paso 2: Especificación de grupos semánticos.

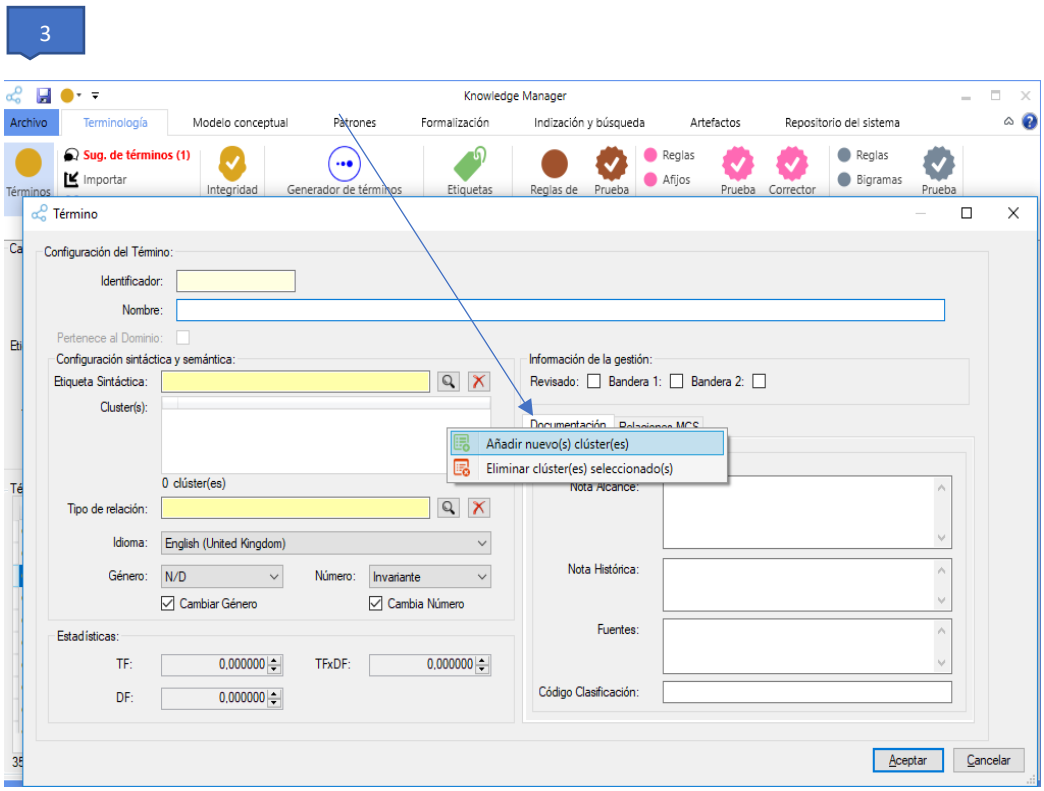


Figura 5. Paso 3: Especificación de grupos semánticos.

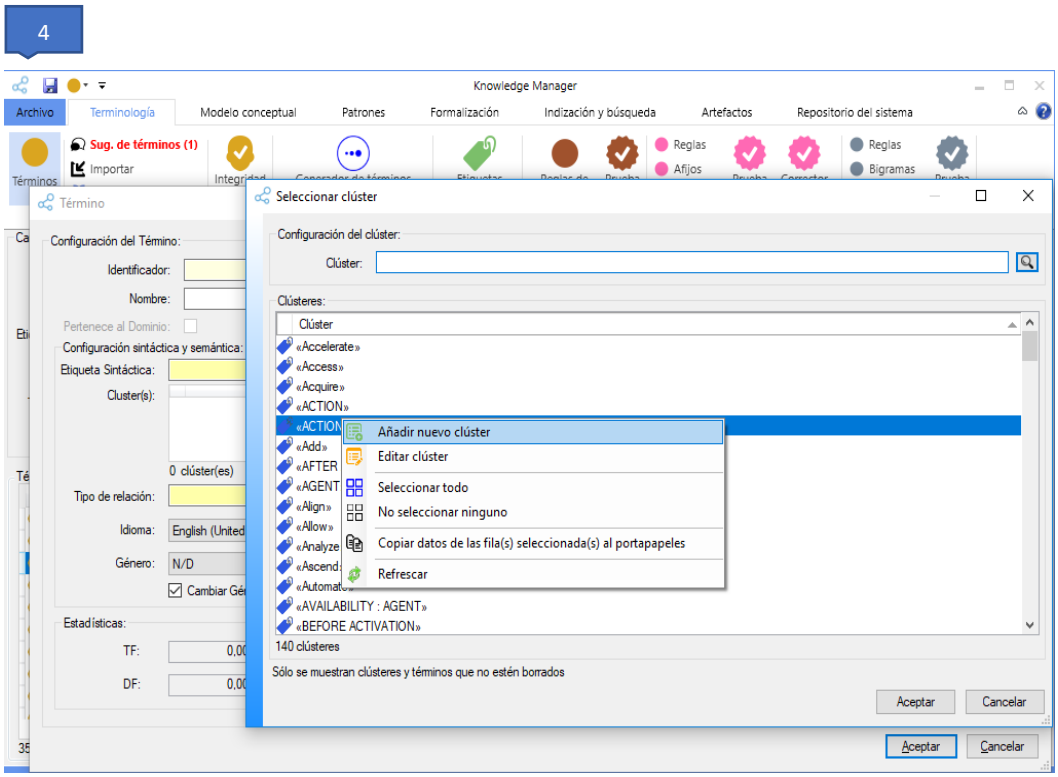


Figura 6. Paso 4: Especificación de grupos semánticos.

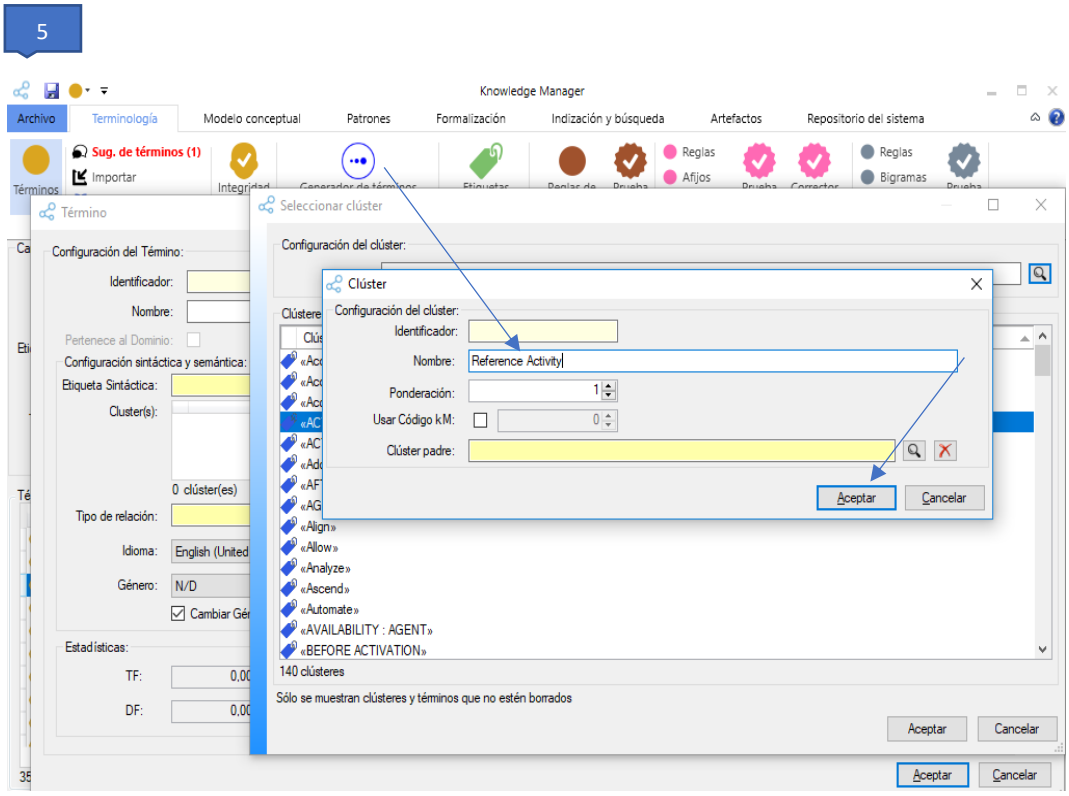


Figura 7. Paso 5: Especificación de grupos semánticos

1.2 Especificación de tipos de relaciones:

KM también apoya la especificación de tipos de relación entre términos. Para representar un estándar de seguridad, se debe crear un tipo de relación para cada asociación en el metamodelo entre las metaclases.

Se introducirán los tipos de relaciones pertenecientes al modelo entre los diferentes tipos de clústeres. Dichas relaciones tendrán la finalidad de asociar los diferentes elementos de un RAF.

Su representación en el metamodelo viene dada por las líneas que unen los rectángulos. Dentro de la herramienta, las relaciones reciben el nombre de "vistas", se tendrá que crear una vista por cada tipo de relación en el modelo.

Los tipos de relaciones son las siguientes:

- Taxonomía (Ya definida por la herramienta).
- PBS (Ya definida por la herramienta).
- Input.
- Output.
- Technique.
- Predecessor.
- Participant.
- Reference Artefact RelationShip.

-Comandos para introducir relaciones:

1.- Click en la sección de la herramienta "Modelo Conceptual".
2.- Click en el icono Tipos de relación.
3.- Click derecho en la tabla "Tipos de relación" y click en "añadir nuevo tipo de relación".
4.- Introducir nombre de la relación y los roles. Los roles son las palabras que aparecen en los extremos de la línea que une los clústeres. De esta forma se irán introduciendo una a una todas las relaciones del modelo.

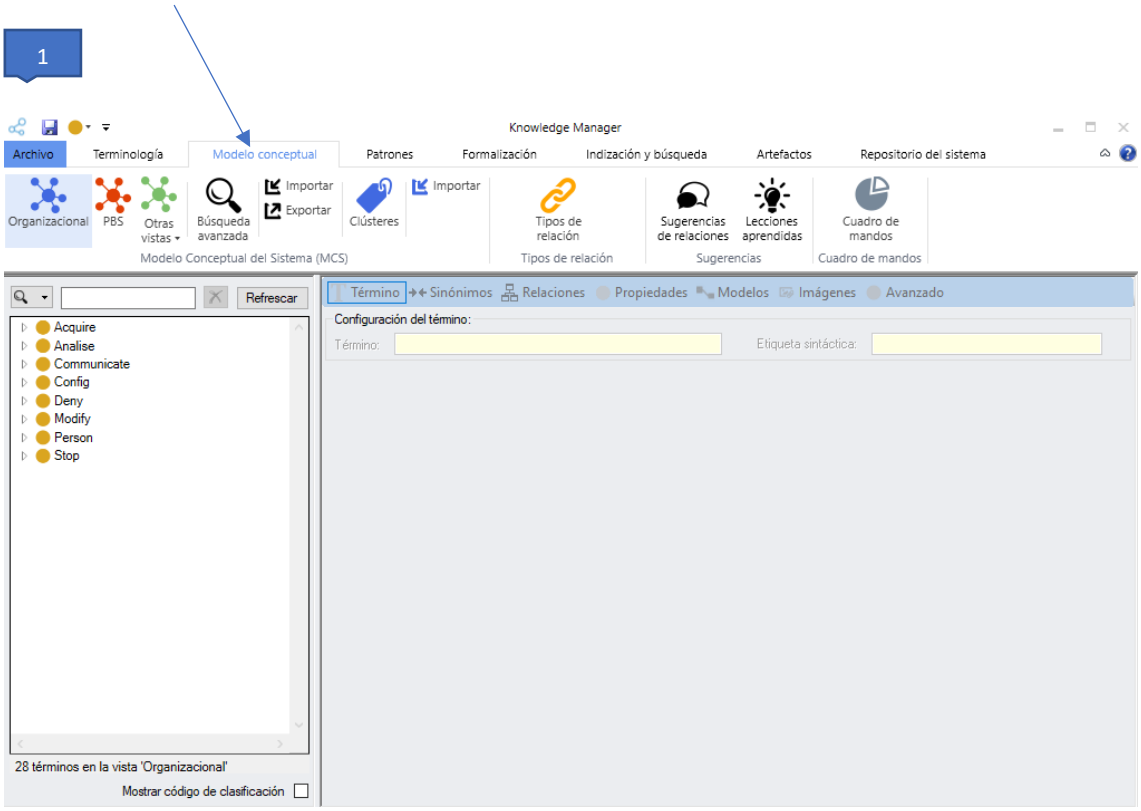


Figura 8. Paso 1: Especificación de tipos de relaciones.

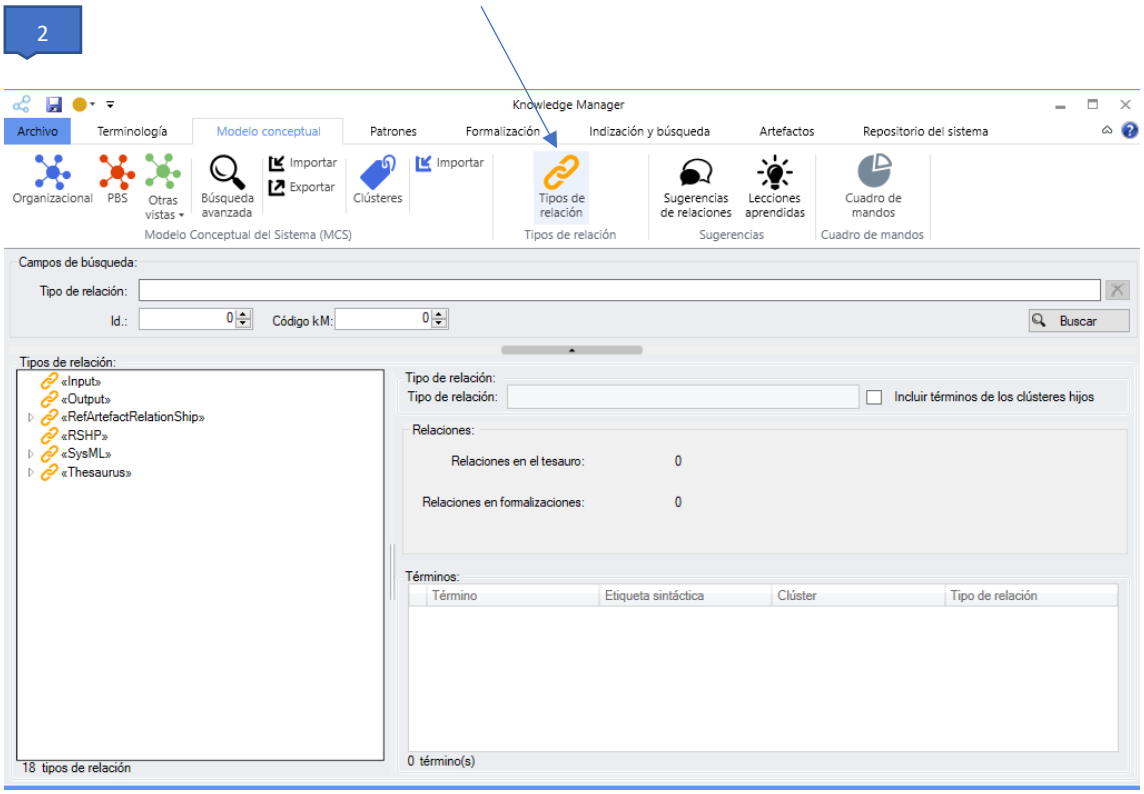


Figura 9. Paso 2: Especificación de tipos de relaciones.

3

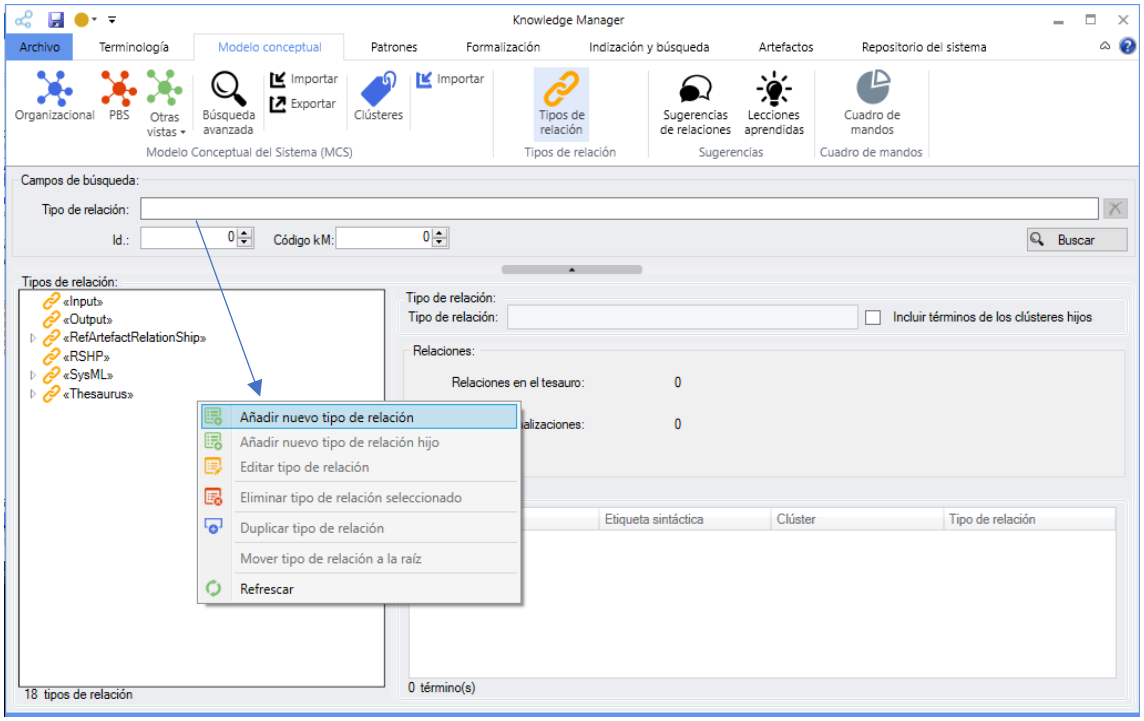


Figura 10. Paso 3: Especificación de tipos de relaciones.

4

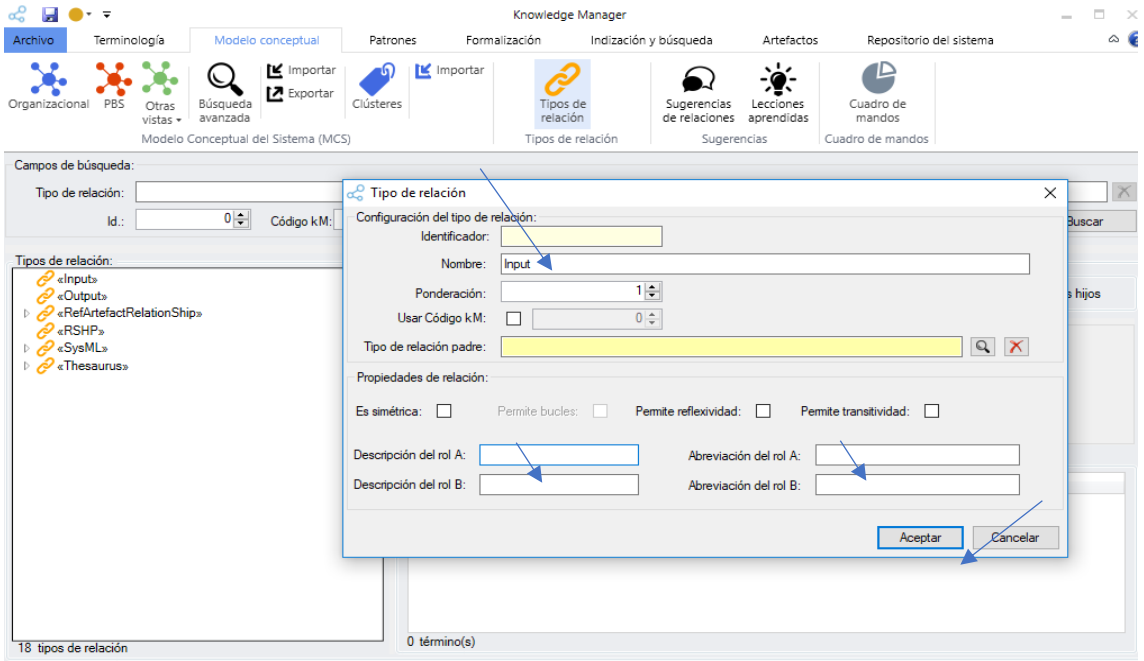


Figura 11. Paso 4: Especificación de tipos de relaciones.

2ª Fase: Especificación de la información de una norma:

Esta actividad resulta de la representación específica de un estándar de seguridad dado. Se pueden distinguir dos tareas. Normalmente, las tareas se ejecutarán iterativamente para representar incrementalmente un estándar de seguridad.

2.1 Especificación de la terminología de un estándar:

En este paso se analizará el texto en busca de términos específicos del estándar. Una vez identificados serán modelados en un clúster u en otro dependiendo de cuál sea su finalidad dentro del estándar.

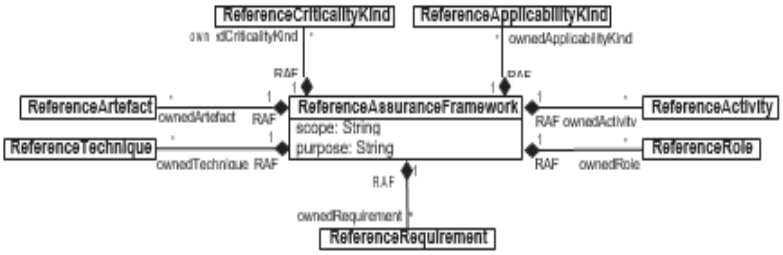


Figura 12. Clústeres del metamodelo

Esta tarea tiene dos aspectos principales a tratar:

1. Términos del glosario: En primer lugar, se analizarán y se clasificarán los términos del glosario con sus respectivos acrónimos, en su clúster correspondiente. Para ello se realizarán dos registros, primero el del acrónimo y después el término completo.

   Cada vez que se agrega un término, es necesario (1) especificar su categoría sintáctica (por ejemplo, sustantivo o acrónimo) y (2) asociarlo con los clústeres a los que pertenece, en este caso en el del propio estándar y en el clúster correspondiente dentro del estándar según su significado semántico.

2. Términos del estándar: A continuación, se debe analizar el texto de la norma para identificar los términos que corresponden al Artefacto de Referencia, el Atributo de Artefacto de Referencia, la Actividad de Referencia, el Rol de Referencia o la Técnica de Referencia.

   Cada vez que se identifica un término, se añade a la Terminología y se asocia el clúster semántico al que pertenezca y al clúster propio del estándar.

-Ejemplo de aplicación:

"The ***software requirement process*** uses the outputs of the ***system life cycle process*** to develop the ***high-level requirements****"*.

En esta frase podemos observar como: *software requirement process, system life cycle process y high-level requirements,* son términos específicos del estándar.

Una vez identificados los términos, el siguiente paso consiste en modelarlos en un clúster del metamodelo. Para ello a través de las definiciones dadas en el párrafo anterior, se obtiene la siguiente conclusión:

- *Software requirement process y system life cycle process:* Son unidades de comportamiento que podría tener que ser ejecutadas, por lo tanto, serian modeladas en el clúster de: "Reference activity".
- *High-level requirements:* Es una unidad de datos que podría tener que ser administrada, por lo tanto, se modelaría como "Reference Artefact".

- Comandos para añadir términos:

1.- Click en la sección de la herramienta "Terminología".
2.- Click derecho dentro de la tabla de términos en la sección de terminología.
3.- A continuación, se desplegarán una serie de opciones. Se dará click a la opción "añadir nuevo término".
4.- Rellenar los campos de nombre y etiqueta sintáctica.
5.- Seleccionar los clústeres al que pertenezca. Cada termino tendrá que estar clasificado por dos clústeres:
    1.- Clúster al que pertenezca el término dentro del metamodelo. Ejemplo: "Reference Artefact".
    2.- Clúster propio de estándar al que el término pertenece. Ejemplo: "DO-178".

Este proceso se repetirá sucesivamente hasta que ya no queden más términos en el standard que analizar.

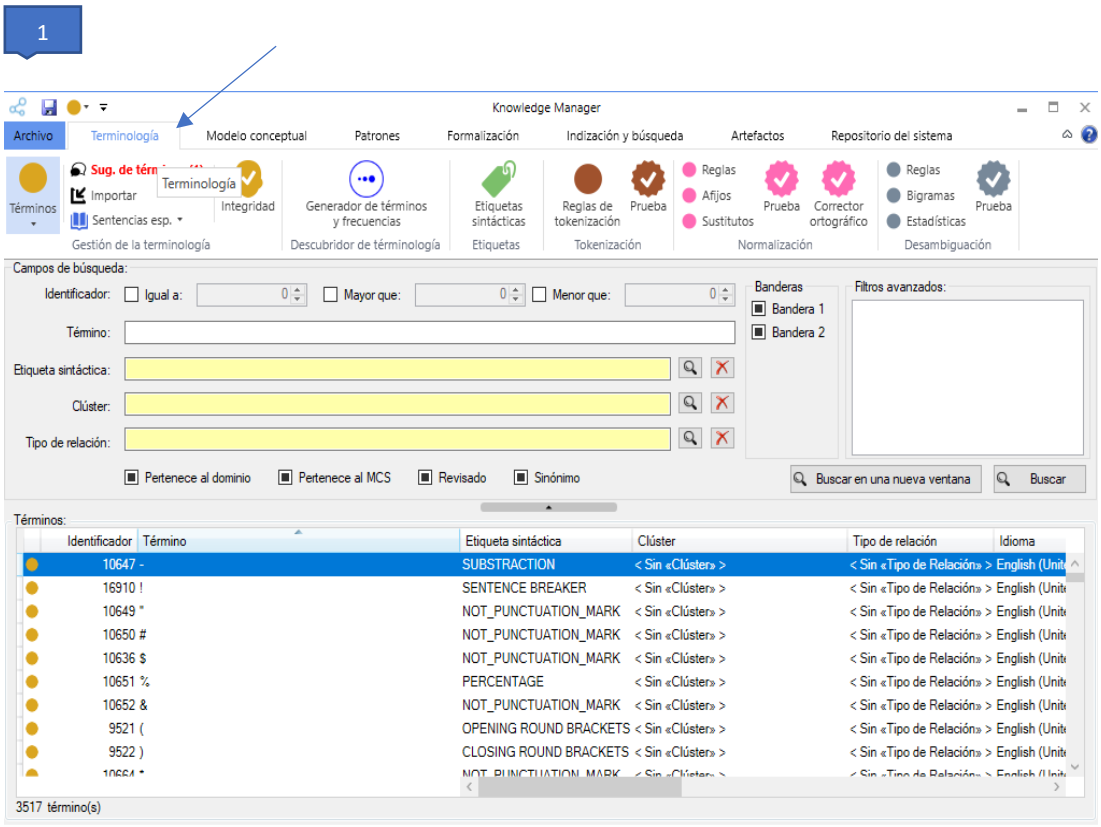


Figura 13. Paso 1: Especificación de la terminología de un estándar.

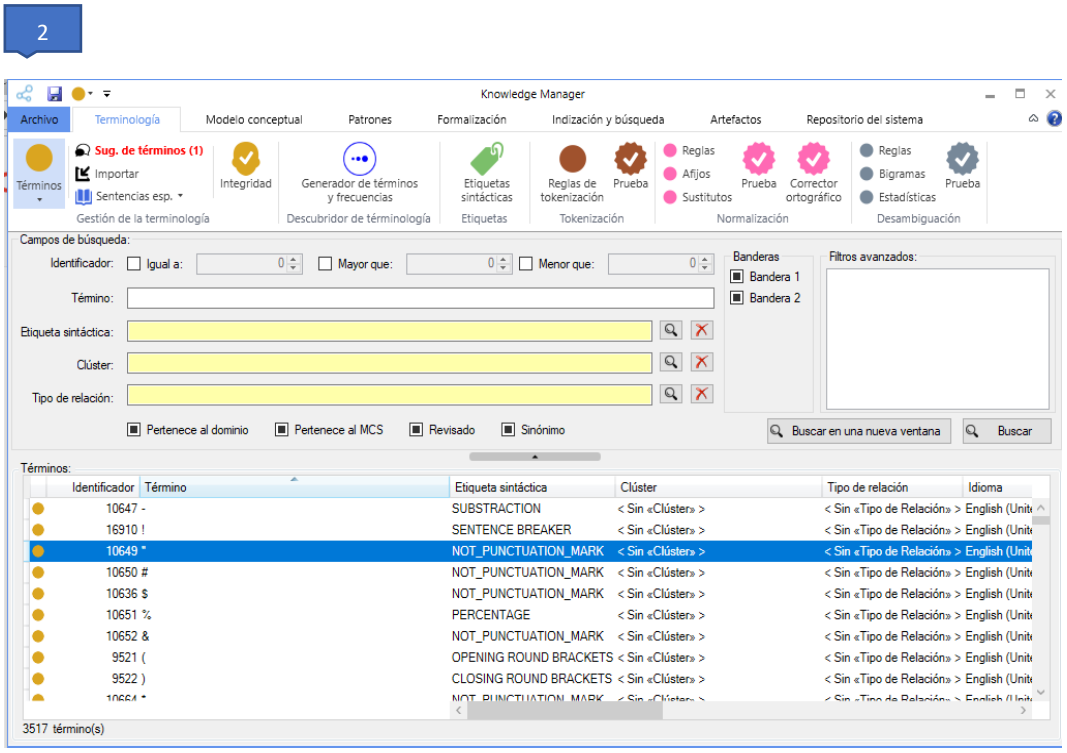


Figura 14. Paso 2: Especificación de la terminología de un estándar.

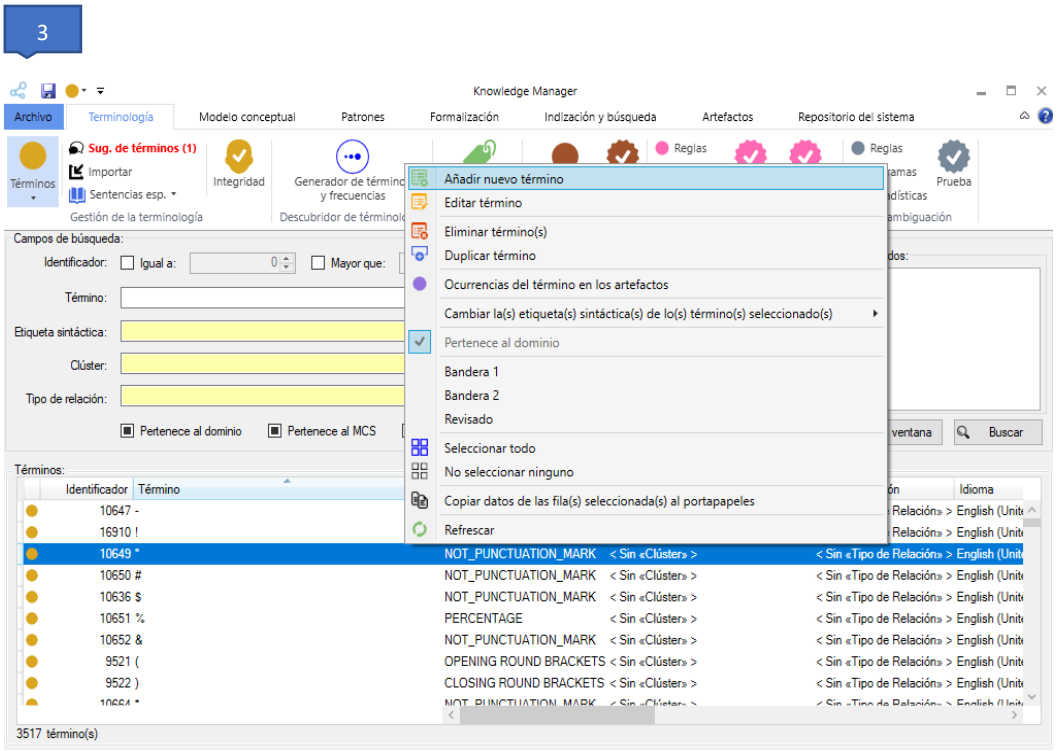


Figura 15. Paso 3: Especificación de la terminología de un estándar.

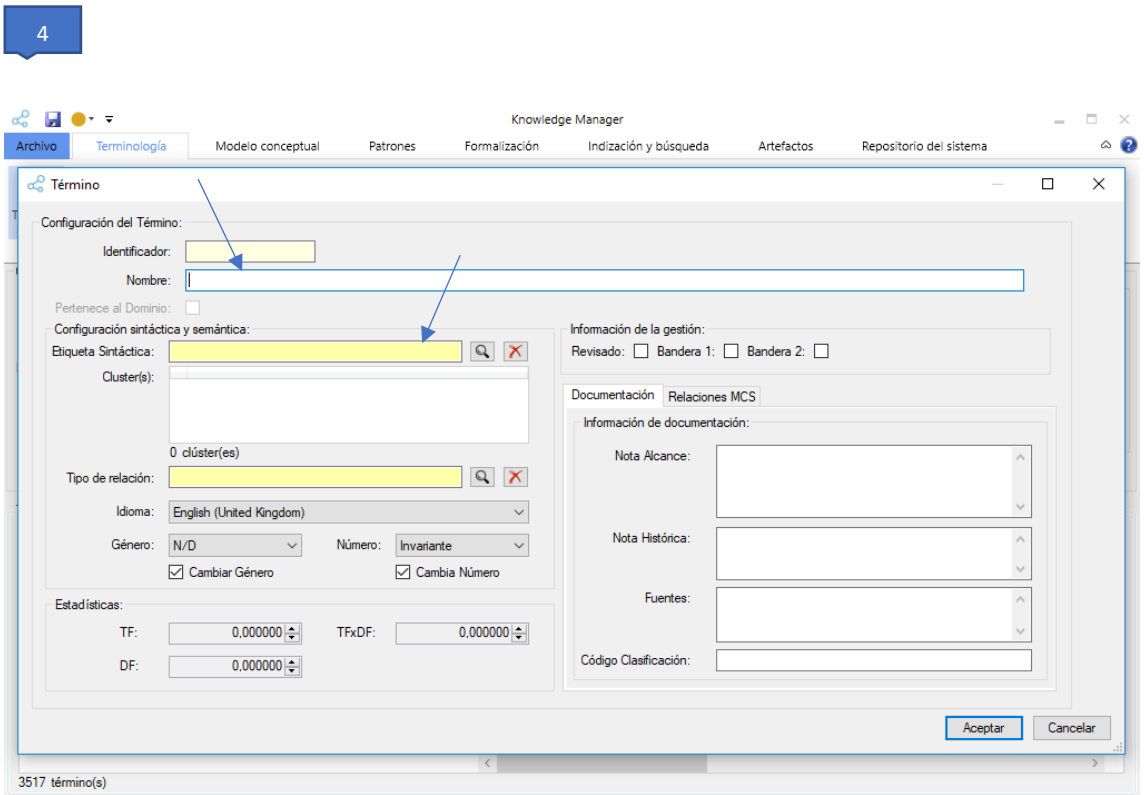


Figura 16. Paso 4: Especificación de la terminología de un estándar.

5

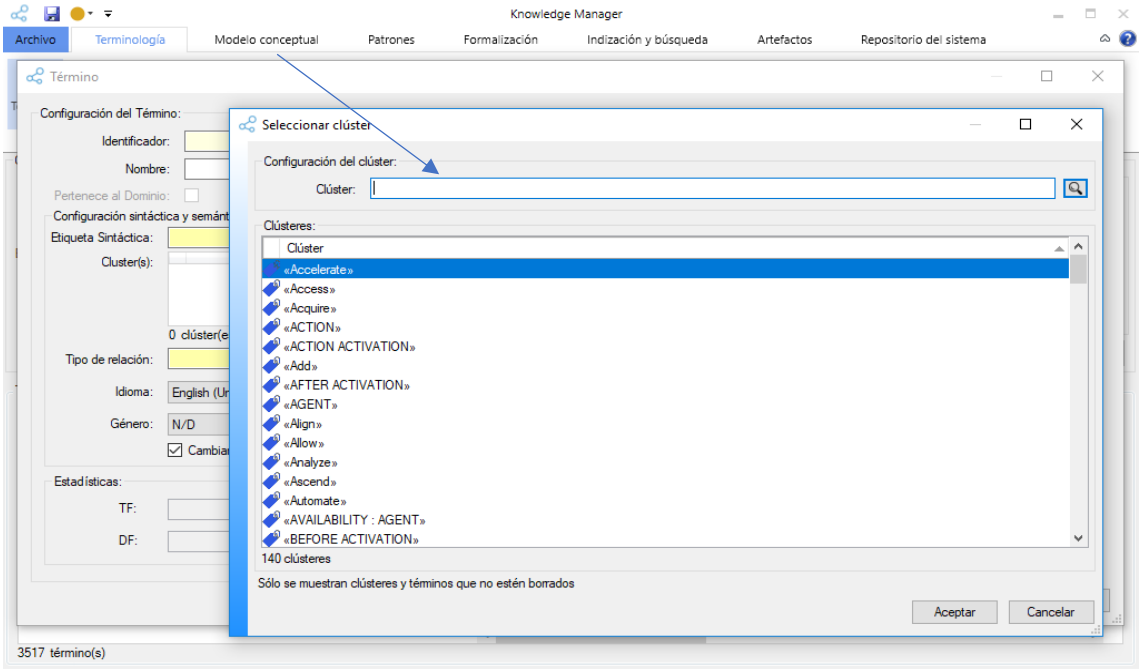


Figura 17. Paso 5: Especificación de la terminología de un estándar.

2.2 Modelar relaciones entre los términos:

Una vez que todos los términos relevantes han sido introducidos y clasificados, las relaciones entre ellos pueden ser especificadas en el modelo Conceptual. Estas relaciones se clasificarán según los tipos de relación disponibles en KM, tanto los predeterminados como los creados durante la configuración de KM.

c

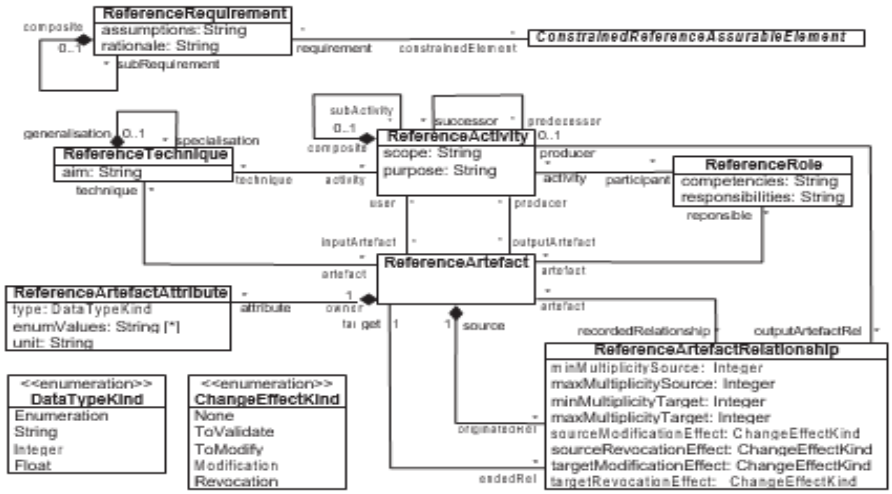


Figura 18. Metamodelo RAF asociaciones.

Un usuario debe ajustarse al metamodelo genérico holístico al especificar relaciones, es decir, sólo los términos que corresponden a los extremos de una asociación dada en el metamodelo deben estar relacionados.

Las relaciones a llevar a cabo serán las citadas en el primer apartado. A continuación, serán descritas una por una:

- Taxonomía: Dicha relación se utilizará a la hora de establecer una taxonomía entre términos.
- Equivalence: Dicha relación se utilizará a la hora de emparejar los acrónimos con su término.
- PBS: Dicha relación se utilizará a la hora de la descomposición de un artefacto o un proceso, es decir, cuáles son los componentes por los que se encuentra formado un artefacto o cuáles son las subactividades por las que se encuentra formado de un proceso.
- Input: Esta relación se utilizará para representar la entrada de artefactos a los diferentes procesos del estándar para la posterior creación de un output.
- Output: Esta relación se utilizará para representar cuales son los artefactos creados por los diferentes procesos del estándar.
- Technique: Dicha relación será utilizada para modelar una técnica especifica dentro de una actividad de referencia. También puede ser aplicada a un artefacto.
- Predecesor: Esta relación sirve para modelar el orden de los procesos del estándar, es decir, cual es el que sucede antes.
- Participant: Dicha relación sirve para asignar el artefacto o el proceso a un rol.
- Reference Artefact Relationship: Esta relación se utiliza para llevar a cabo las relaciones entre artefactos. Suele estar definida por el "verbo" que acompaña a ambos artefactos en el standard. Ej.: conform to, verifies, base on…

El usuario también debe decidir si las relaciones entre Artefactos de Referencia deben especificarse como especializaciones, como composiciones o con el tipo de Relación Artefacto de Referencia.

También es posible definir especializaciones de este tipo de relación si un usuario lo decide, por ejemplo, porque es una Relación de Artefacto de Referencia recurrente. Por ejemplo, es común que los artefactos tengan que ser "conforme" a algún plan o estándar.

Finalmente, también puede ser necesario especificar relaciones de especialización y equivalencia entre términos; Por ejemplo, 'MC / DC' y 'Modified Condition / Decision Coverage' son equivalentes para DO-178C.

-*Ejemplo de aplicación:*

"The ***software requirement process*** uses the outputs of the ***system life cycle process*** to develop the ***high-level requirements***".

En esta frase podemos observar a través del contexto y el metamodelo como:

- *System life cycle process* es una actividad predecesora de *software requirement process.* Por lo tanto, estos términos se encuentran relacionados a través de la vista predecesor.
- *High-level requierements* es un artefacto producido por la *actividad software requirement process.* Por lo tanto, estos términos se encuentran relacionados por la vista output.

El proceso de establecer relaciones lo dividiremos en tres partes:

1. Modelado de Acrónimos: En esta parte inicial se establecerán las relaciones entre los acrónimos con sus respectivos términos del glosario. La relación a utilizar será "Equivalence".

2. Modelado de Proceso: En esta parte se establecerán las relaciones de los términos relacionados con el proceso. Los términos participantes serán los Reference Activity y Reference Artefact. Las relaciones a llevar a cabo serán únicamente de output, input, subactivity (PBS) y predecesor.

   El objetivo de esta parte consiste básicamente en modelar cuales son los procesos que se preceden, cuáles son las actividades por las que se encuentran formados y cuáles son sus entradas y salidas.

3. Modelado de Artefacto: En esta parte se establecerán las relaciones entre los términos relacionados con los artefactos. Los términos participantes serán los Reference Artefact. Las relaciones a llevar a cabo serán Taxonomía, PBS y Reference Artefact Relationship.

El objetivo de dicha parte consiste en modelar las relaciones entre los diferentes artefactos del estándar y su composición.

-*Comandos para establecer relaciones:*

1.- Click en la sección de la herramienta "Modelo Conceptual".
2.- Click en el tipo de vista que queramos introducir.
3.- Click derecho en la tabla en blanco por defecto y aparecerán una serie opciones.
4.- Dentro de las opciones, si queremos crear un término padre de la relación clicaremos en la opción "añadir nuevo término o relación" y si queremos añadir un término a una relación se dará click derecho sobre el termino padre y después en "añadir un nuevo hijo".

Este proceso se repetirá sucesivamente hasta que ya no queden más relaciones en el standard que analizar e identificar.

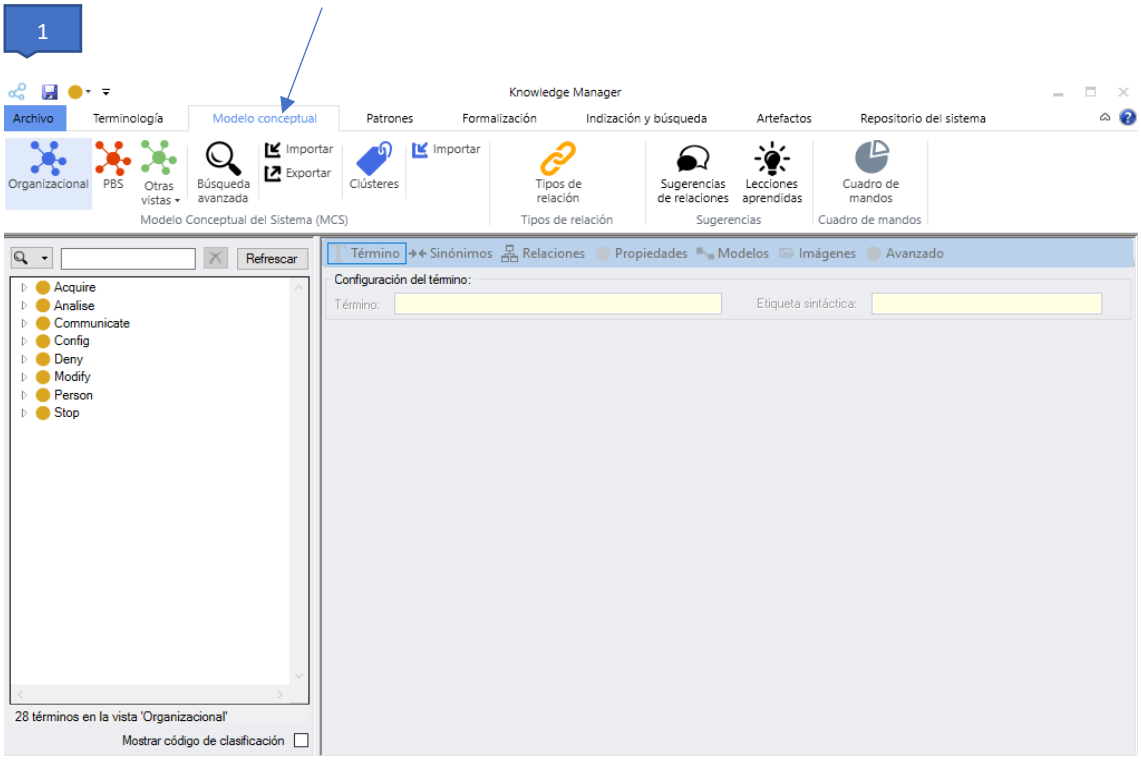


Figura 19. Paso 1: Modelar relaciones entre los términos.

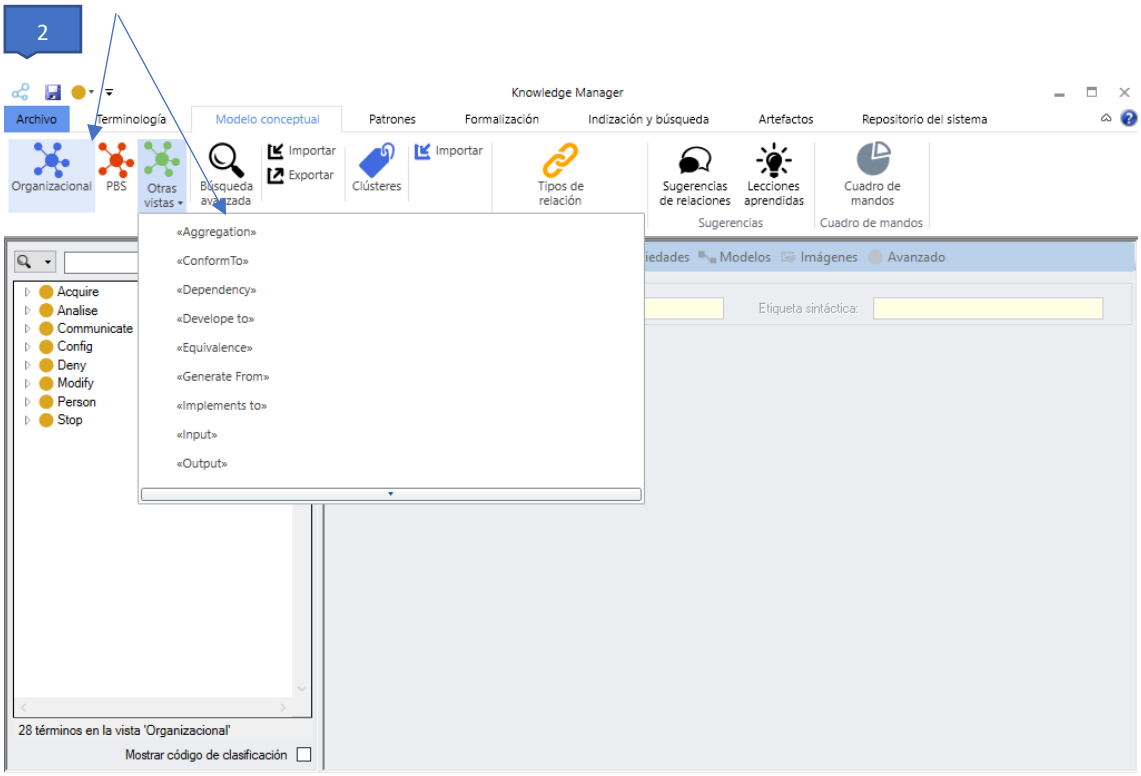


Figura 20. Paso 2: Modelar relaciones entre los términos.

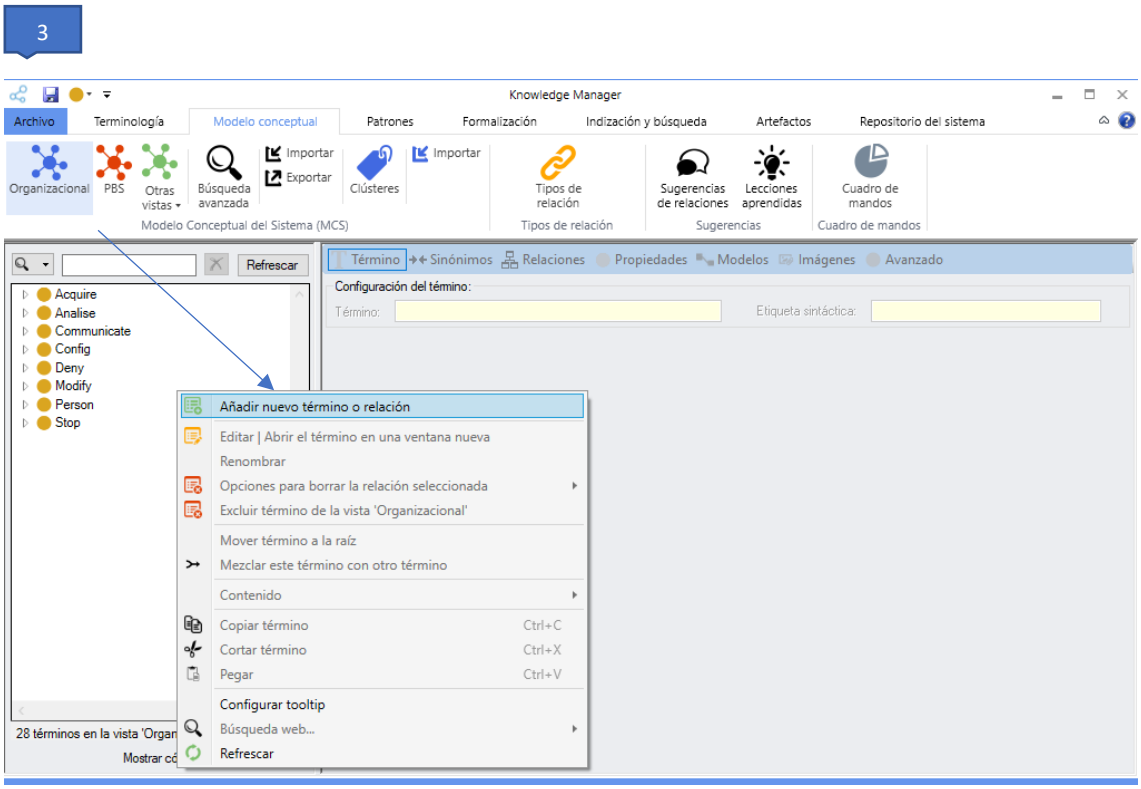


Figura 21. Paso 3: Modelar relaciones entre los términos.

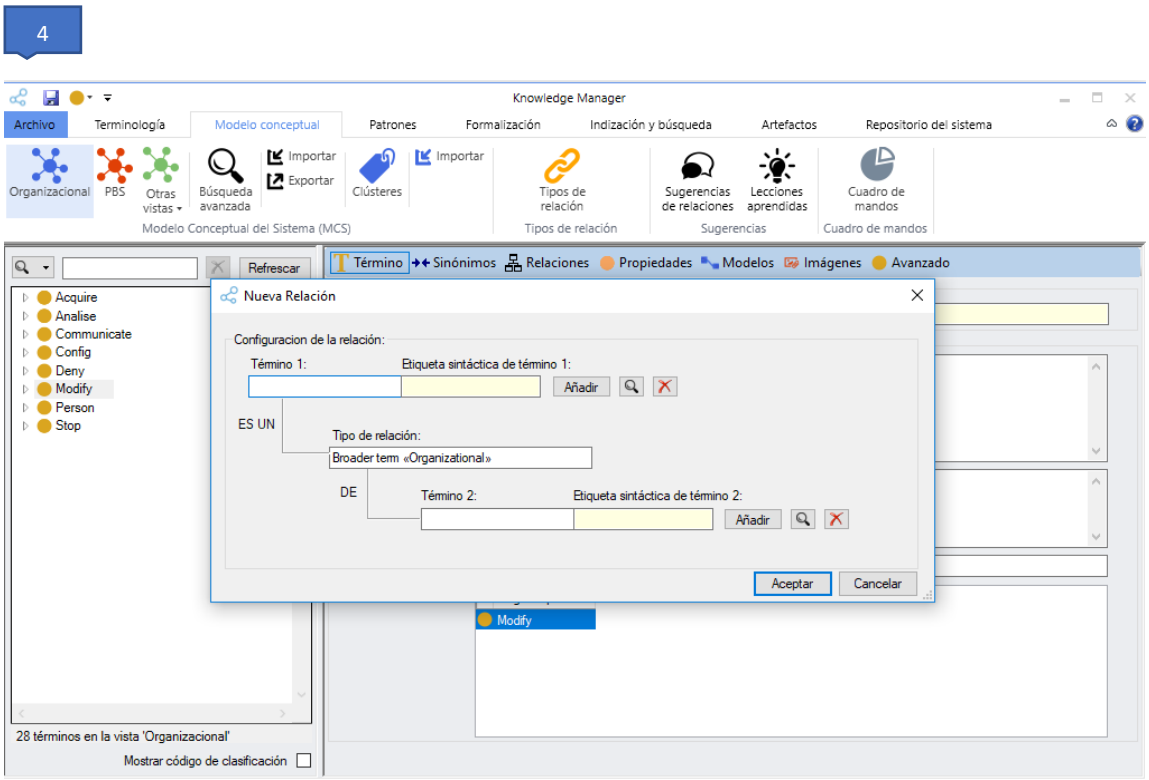


Figura 22. Paso 4: Modelar relaciones entre los términos; creación de una relación término padre.

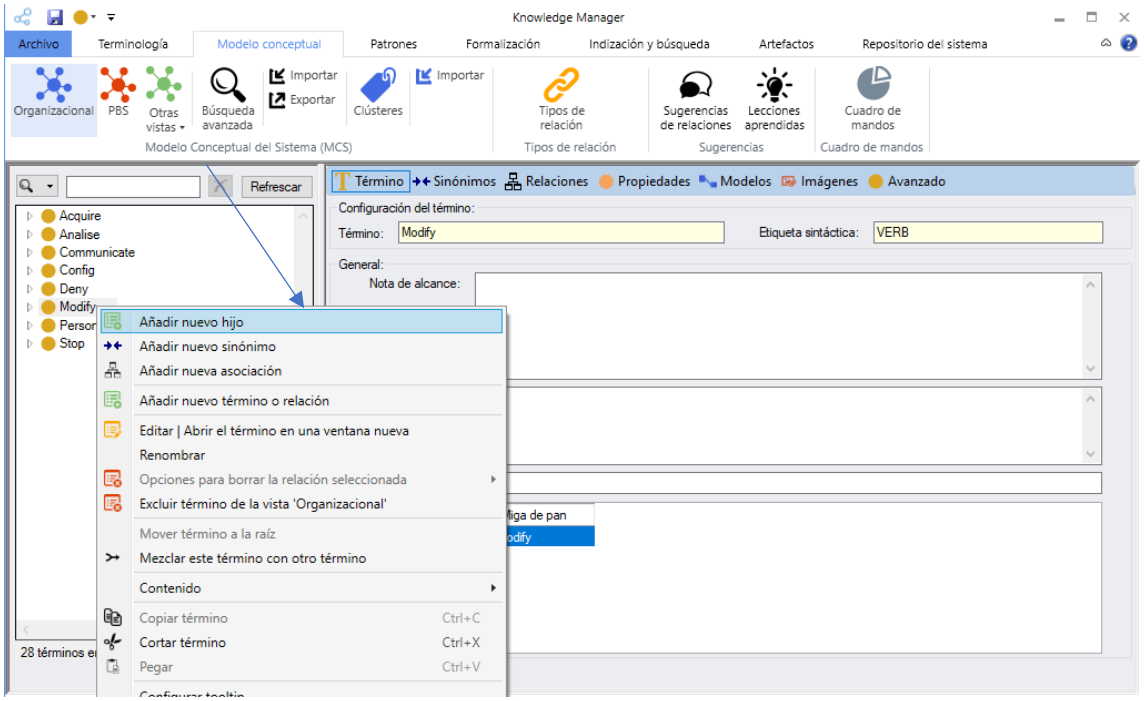


Figura 23. Paso 4: Modelar relaciones entre los términos; creación de una relación término hijo.

3.5.- Discusión:

Una vez descrita la propuesta, esta sección discute cómo la representación de un estándar de seguridad con KM puede ser explotada para fines específicos de seguridad y certificación.

Dentro del propósito general de demostrar la alineación con el cumplimiento de una norma de seguridad, contemplamos seis posibilidades principales para aprovechar las representaciones:

a) Análisis de la calidad del texto de una norma de seguridad

KM es parte de un conjunto de herramientas que soporta, entre otras características, análisis de la calidad de artefactos de sistema, incluyendo artefactos textuales.

Más concretamente, la suite puede analizar la corrección de los artefactos, la completitud y la consistencia. Teniendo en cuenta el texto de un estándar de seguridad como un ejemplo de artefacto, su calidad de texto podría ser determinada.

Esto sería valioso porque la calidad del texto es una de las debilidades más frecuentes que los profesionales encuentran en los estándares de seguridad. Podrían identificarse las partes que podrían especificarse mejor o deberían aclararse.

b) La alineación especificación del sistema.

Al especificar información para un sistema específico o analizar la información, se podría evaluar el grado en que la especificación se alinea con un estándar dado.

En primer lugar, para el sistema podrían especificarse, por ejemplo, sus requisitos del sistema, de acuerdo con patrones que se refieren a los clústeres semánticos agregados o a los términos específicos del estándar.

En segundo lugar, una ontología del sistema podría estar vinculada a la ontología del estándar, por ejemplo, para especificar que una parte dada del sistema corresponde al concepto de componente DO-178C.

c) Evaluación del cumplimiento.

Se podría utilizar una ontología de un estándar de seguridad creada con KM para evaluar el cumplimiento de procesos y productos.

Las capacidades de la suite de herramientas podrían utilizarse para comparar información de proceso o producto con la ontología, con el fin de determinar las lagunas de cumplimiento.

La información podría corresponder a artefactos de diferente naturaleza: especificaciones textuales, documentos, diagramas, hojas de cálculo ...

d) Comparación de normas.

El texto u ontología de un estándar de seguridad podría ser comparado con la ontología de otro, con el fin de identificar las similitudes y las diferencias.

e) La reutilización de la información de un sistema.

Si la información de un sistema (por ejemplo, un modelo de sistema) se vincula con la ontología de una norma de seguridad para declarar el cumplimiento de la norma, sería posible buscar información del sistema compatible y, cuando se encuentre, volverla a utilizar.

Incluso podría ser posible analizar la reutilización de información del sistema entre normas de seguridad si las ontologías de las diferentes normas están vinculadas. La vinculación de la información de un sistema con la ontología podría basarse en (b).

f) Especificación de métricas específicas a un estándar.

Se podrían diseñar métricas específicas dentro de la capa de reglas de inferencia basada en la información semántica de un estándar de seguridad representado en KM.

Las métricas podrían evaluar (1) el cumplimiento general de la norma (por ejemplo, la cantidad de artefactos de referencia que se han proporcionado) y (2) las características específicas del artefacto que define una norma (por ejemplo, la coherencia de la especificación de la arquitectura).

Aunque las métricas a menudo no se declaran directamente en el estándar de seguridad (por ejemplo, para el último ejemplo), la información de los estándares conduciría a su definición indicando las áreas para las cuales se podrían diseñar métricas y los posibles aspectos a considerar.

La forma en que estas posibilidades pueden finalmente materializarse es parte de nuestro trabajo actual y futuro, que podría incluir la explotación de otros beneficios.

4.- APLICACIÓN Y VALIDACIÓN DE LA APROXIMACIÓN:

En este apartado se aplicará y se validará la aproximación presentada en la sección anterior a dos casos reales. Los dos estándares elegidos son los siguientes:

- Estándar DO-178C: Estándar para el desarrollo de software en el sector de seguridad crítica de la aviación.
- Estándar EN-50128: Norma que rige el desarrollo de software en el sector ferroviario.

En primer lugar, se realiza su representación ontológica a través de la herramienta KM, para después validar y analizar el resultado de la aplicación a través de la herramienta RQA, con la que estudiaremos la calidad de los estándares y la posible existencia de ambigüedades y contradicciones dentro de ellos.

Se finaliza con una discusión en la que se llevan a cabo las conclusiones y aspectos a mejorar dentro del estándar.

4.1.- Caso 1: DO-178C:

En este apartado se realiza lo anteriormente explicado en una subsección del Estándar DO-178C. La razón por la que ha sido seleccionada esta subsección ha sido porque existen modelos disponibles de este estándar validado por profesionales.

4.1.1.- Representación del estándar:

En este apartado se lleva a cabo la representación ontológica del estándar DO-178C. Para ello se utiliza la metodología explicada en el punto 3 de este TFG.

La metodología explicada consta de dos fases a través de las cuales se realiza la representación:

1ª Fase: Configuración KM:

En esta primera fase se realizará la configuración de la herramienta KM en base al estándar analizado. Se deben realizar dos tareas:

1.1 Especificación de grupos semánticos:

En esta tarea los clústeres del estándar DO-178C deben agregarse a la capa del modelo conceptual para poder indicar el tipo de información que representa un término.

En primer lugar, es necesario un clúster con el nombre del estándar de seguridad que se va a representar para especificar más adelante que un término cae dentro del alcance de la norma.

En segundo lugar, se introducirán los tipos de clúster pertenecientes al metamodelo. Estos clústeres tendrán la finalidad de modelar los diferentes elementos de una RAF.

-Aplicación:

1. El primer paso consiste en crear un clúster con el nombre del estándar, para ello seguiremos los pasos establecidos por la metodología para crear un clúster, punto 3.4.

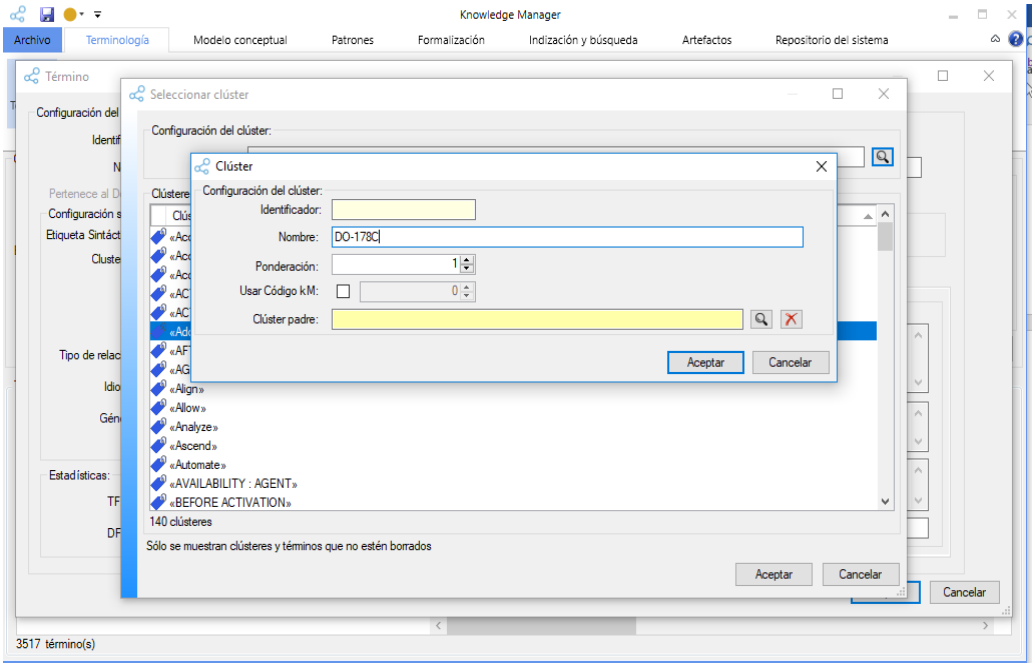


Figura 24. Creación del clúster propio del estándar.

2. Una vez creado el clúster del estándar se pasará a crear los clústeres del metamodelo holístico presentado en el punto 3.2.

   En el caso del estándar DO-178C los clústeres que aparecen son los siguientes:

   - Reference Artefact: El clúster "Reference Artefact" hace referencia a las unidades de datos que podrían tener que ser administradas dentro del estándar DO-178C.

     En este caso el clúster Reference Artefact se tendría que configurar en la herramienta KM puesto que aparecen términos que estarían asociados a este.

Para crear el clúster se seguirán los pasos estipulados por la metodología explicada en el punto 3.4.

Algunos ejemplos dentro del estándar que se corresponderían con este clúster serían: “Derived High-Level Requirements”, “Software Requirements Data”, “High-level Requirements”, “Software requirements Standards” …

- Reference Activity: El clúster “Reference Activity” hace referencia a las unidades de comportamiento que podría tener que ser ejecutadas dentro del estándar DO-178C.

  En este caso el clúster Reference Activity se tendría que configurar en la herramienta KM puesto que aparecen términos que estarían asociados a este.

  Para crear el clúster se seguirán los pasos estipulados por la metodología explicada en el punto 3.4

  Algunos ejemplos dentro del estándar que se corresponderían con este clúster serían: “System Life Cycle”, “Software Planning process”, “Software Development Plan”, “Software Requirement Process”…

- Reference Attribute: El clúster “Reference Atributte” hace referencia a las características de un artefacto de referencia que podrían ser registradas dentro del estándar DO-178C.

  En este caso el clúster Reference Atributte se tendría que configurar en la herramienta KM puesto que aparecen términos que estarían asociados a este.

  Para crear el clúster se seguirán los pasos estipulados por la metodología explicada en el punto 3.4

  Algunos ejemplos dentro del estándar que se corresponderían con este clúster serían: “Rationale” y “Robustness”.

- Reference Role: El clúster “Reference Role” hace referencia a los tipos de agentes que podrían tener que estar involucrados dentro del estándar DO-178C.

En este caso el clúster Reference Role se tendría que configurar en la herramienta KM puesto que aparecen términos que estarían asociados a este.

Para crear el clúster se seguirán los pasos estipulados por la metodología explicada en el punto 3.4

Algunos ejemplos dentro del estándar que se corresponderían con este clúster serían: “Certification Authorities Software Team” y “Working Group”.

- Reference Technique: El clúster “Reference Technique” hace referencia a a las formas específicas de ejecutar una actividad de referencia o crear un artefacto de referencia dentro estándar DO-178C.

En este caso el clúster Reference Technique se tendría que configurar en la herramienta KM puesto que aparecen términos que estarían asociados a este.

Para crear el clúster se seguirán los pasos estipulados por la metodología explicada en el punto 3.4

Algunos ejemplos dentro del estándar que se corresponderían con este clúster serían: “Emulator”, “Autocode generator” y “Simulator”.

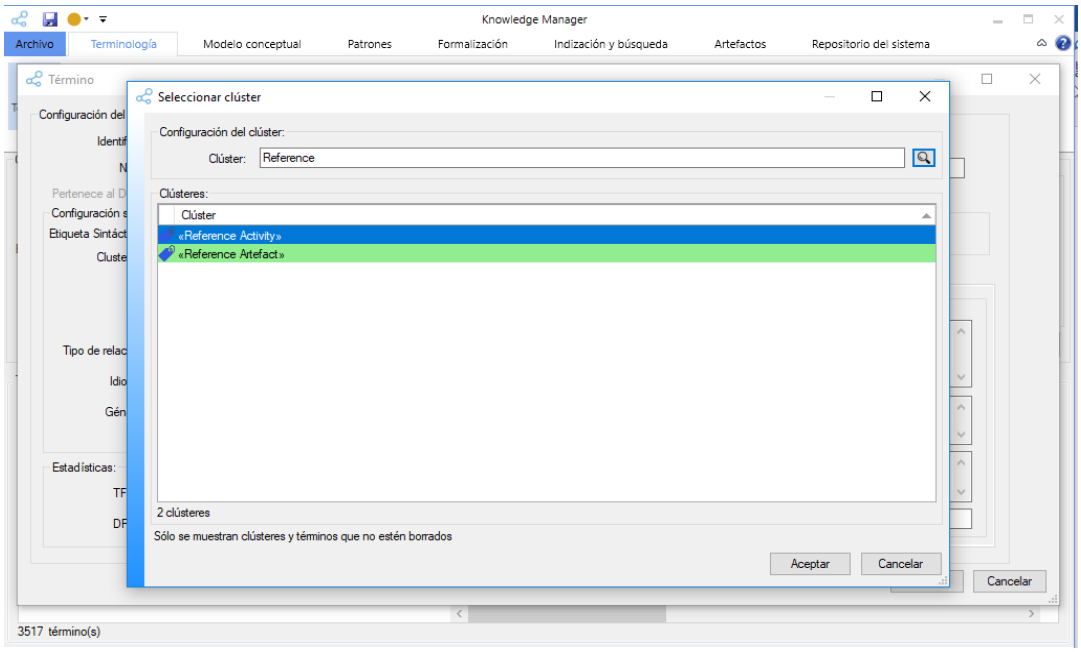


Figura 25. Creación de los clústeres del metamodelo pertenecientes al estándar.

1.2 Especificación de tipos de relaciones:

En esta segunda tarea se introducirán los tipos de relaciones pertenecientes al metamodelo (Punto 3.2) entre los diferentes tipos de clústeres. Dichas relaciones tendrán la finalidad de asociar los diferentes elementos del estándar DO-178C.

-Aplicación:

Las relaciones que configurar dentro de este estándar son las siguientes:

- Taxonomía (Ya definida por la herramienta): Se define esta relación para representar las posibles taxonomías entre términos del estándar. Esta relación se lleva a cabo entre términos pertenecientes al clúster Reference Artefact.
- Equivalence (Ya definida por la herramienta): Se define esta relación para asociar los términos del estándar con sus respectivos acrónimos. Esta relación se lleva a cabo entre términos pertenecientes al clúster Reference Artefact.
- PBS (Ya definida por la herramienta): Se define esta relación para representar la composición entre término. Esta relación se lleva a cabo entre términos pertenecientes al clúster Reference Artefact y Reference Activity.
- Input: Se define esta relación para representar la entrada de las actividades propias del estándar. Esta relación se lleva a cabo entre términos pertenecientes al clúster Reference Artefact - Reference Activity.
- Output: Se define esta relación para representar la salida de las actividades propias del estándar. Esta relación se lleva a cabo entre términos pertenecientes al clúster Reference Artefact - Reference Activity.
- Predecesor: Se define esta relación para representar el orden las actividades propias del estándar. Esta relación se lleva a cabo entre términos pertenecientes al clúster Reference Activity.
- Reference Artefact RelationShip: Se define esta relación para representar la relación existente entre los términos pertenecientes al clúster Reference Artefact.

Dichas relaciones serán configuradas en la herramienta KM según la metodología explicada en el punto 3.4.

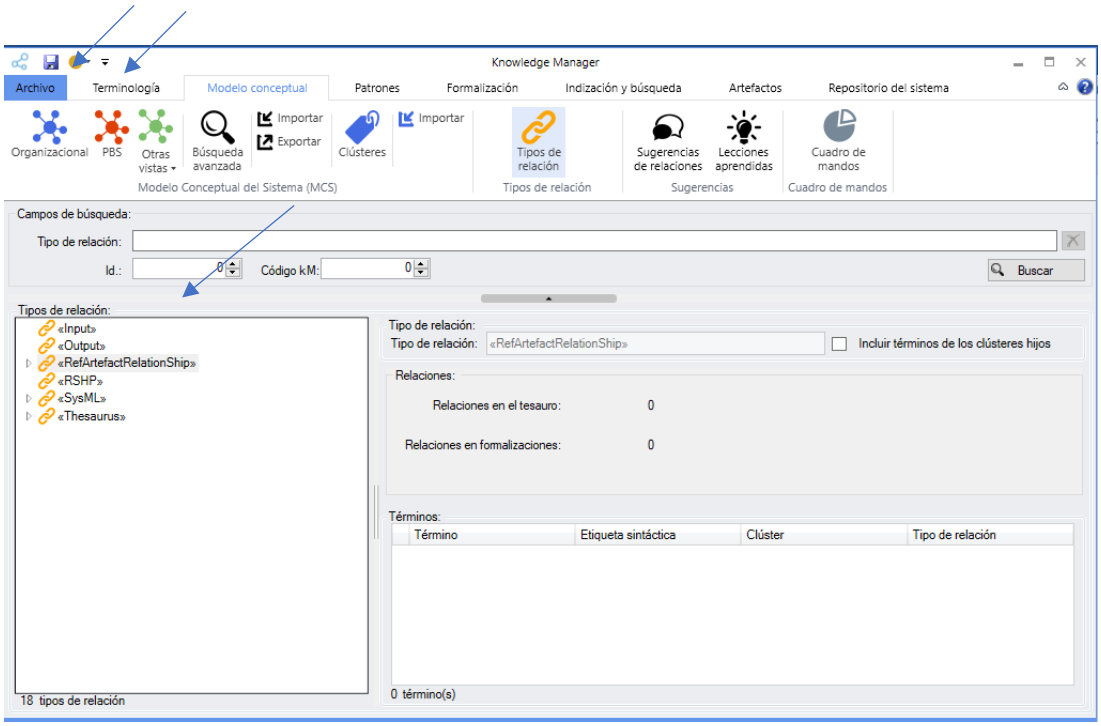


Figura 26. Configuración de las relaciones del metamodelo pertenecientes al estándar.

2ª Fase: Especificación de la información de una norma:

Una vez configurada la herramienta KM en base al estándar D-178C, el siguiente paso consiste en la especificación de la información en base al metamodelo del punto 3.2, para ser después modelada en la herramienta.

En esta fase se distinguen dos tareas. Normalmente, las tareas se ejecutarán iterativamente para representar incrementalmente el estándar de seguridad.

2.1 Especificación de la terminología de un estándar:

En esta primera actividad se analizará el texto en busca de términos específicos del estándar DO-178C. Una vez identificados serán modelados en un clúster u en otro dependiendo de cuál sea su finalidad dentro del estándar.

Un término pertenecerá a un clúster o a otro dependiendo de sus características dentro del contexto del estándar. Las características vienen dadas en el punto 3.4 de la metodología.

-Aplicación:

Esta tarea tiene dos aspectos principales a tratar:

1. Términos glosario: En primer lugar, se analizarán y se clasificarán los términos del glosario con sus respectivos acrónimos, en su clúster correspondiente.

   El glosario dentro del estándar DO-178C se encuentra en el anexo B; "Acronyms and Glossary of terms", página 107.

   Dentro el estándar DO-178C los términos del glosario encontrados con sus respectivos clústeres han sido los siguientes:

Acrónimo	Significado	Clúster
ARP	Aerospace Recommended Practice	-
ATM	Air Traffic Management	-
CAST	Certification Authorities Software Team	Reference Role
CNS	Communication, Navigation and Surveillance	-
COTS	Commercial-Off-The-Shelf	-
CRC	Cyclic Redundancy Check	Reference Activity
DO	Document	Reference Artefact
EASA	European Aviation Safety Agency	Reference Role
EUROCAE	European Organization for Civil Aviation Equipment	Reference Role
FAA	Federal Aviation Administration	Reference Role
I/O	Input/Output	-
MC/DC	Modified Condition/Decision Coverage	Reference Techinique
PMC	Program Management Committee	Reference Role
PSAC	Plan for Software Aspects of Certification	Reference Artefact
SAE	Society of Automotive Engineers	Reference Role
SC	Special Committee	Reference Role
SCI	Software Configurativo Index	Reference Artefact
SCM	Software configuration management	Reference Activity

SDP	Software development Plan	Reference Artefact
SECI	Software Life Cycle Enviroment Index	Reference Artefact
SQA	Software Quality Assurance	Reference Artefact
SQA	Software Quality Assurance	Reference Activity
SVP	Software Verification Plan	Reference Artefact
TOR	Terms of Reference	Reference Artefact
TQL	Tool Qualification Level	-
WG	Working group	Reference Role

Tabla 1. Acrónimos del estándar DO-178C

Una vez identificados y clasificados los términos del glosario, el siguiente paso consiste en registrar tanto el acrónimo como el término en la herramienta KM con su clúster correspondiente, para ello seguiremos los pasos estipulados en el punto 3.4 de la metodología.

2. Términos del estándar: En segundo lugar, se analizará el texto del estándar DO-178C con el objetivo de identificar los términos a representar en la aproximación.

Dentro el estándar DO-178C los términos encontrados con sus respectivos clústeres han sido los siguientes:

Número	Término	Clúster
#1	Aeronautical data	Reference Artefact
#2	Airborne	-
#3	Alternative method	-
#4	Anomalous behavior	Reference Attribute
#5	Applicant	Reference Role
#6	Approval	-
#7	Aproved Source	Reference Artefact
#8	Assurance	Reference Attribute
#9	Audit	Reference Activity
#10	Autocode generator	Reference Technique
#11	Baseline	Reference Artefact

#12	Certification	Reference Artefact
#13	Certification authority	Reference Role
#14	Certification credit	Reference Artefact
#15	Change control	-
#16	Check the input of Software Process	Reference Activity
#17	Compiler	Reference Artefact
#18	Component	Reference Artefact
#19	Condition	-
#20	Configuration identification	Reference Attribute
#21	Configuration item	Reference Artefact
#22	Configuration management	Reference Activity
#23	Configuration status accounting	Reference Activity
#24	Control category	-
#25	Control coupling	Reference Activity
#26	Control program	Reference Activity
#27	Data coupling	Reference Artefact
#28	Data dictionary	Reference Artefact
#29	Database	Reference Artefact
#30	Dead code	Reference Artefact
#31	Derive High-level Requirement	Reference Artefact
#32	Design description	Reference Artefact
#33	Embedded identifier	Reference Artefact
#34	Emulator	Reference Technique
#35	Equivalence class	Reference Technique
#36	Equivalent safety	Reference Artefact
#37	Executable Object Code	Reference Artefact
#38	Failure	-
#39	Failure condition	-
#40	Fault	-
#41	Hardware interface	Reference Artefact
#42	High-level Requirements	Reference Artefact
#43	Host computer	Reference Artefact

#44	Integration process	Reference Activity
#45	Low-Level Requirements	Reference Artefact
#46	Memory device	Reference Artefact
#47	Monitoring	Reference Activity
#48	Object Code	Reference Artefact
#49	Parameter data item	Reference Artefact
#50	Parameter Data Item File	Reference Artefact
#51	Patch	Reference Artefact
#52	Previously developed software	Reference Artefact
#53	Product service history	Reference Artefact
#54	Robustness	Reference Attribute
#55	Safety monitoring	Reference Activity
#56	Simulator	Reference Technique
#57	Software Achitecture	Reference Artefact
#58	Software architecture	Reference Artefact
#59	Software assurance	Reference Artefact
#60	Software change	Reference Artefact
#61	Software Coding Process	Reference Activity
#62	Software conformity review	Reference Activity
#63	Software design process	Reference Activity
#64	Software Development Plan	Reference Artefact
#65	Software Development process traceability	Reference Activity
#66	Software development standards	Reference Activity
#67	Software integration	Reference Activity
#68	Software life cycle	Reference Activity
#69	Software requirement	Reference Artefact
#70	Software Requirement Data	Reference Artefact
#71	Software Requirement Data	Reference Artefact
#72	Software Requirement Process	Reference Activity
#73	Software Requirements Standards	Reference Artefact
#74	Software Requirements Standards	Reference Artefact

#75	Source Code	Reference Artefact
#76	Statement coverage	Reference Technique
#77	Structural coverage analysis	Reference Artefact
#78	System Architecture	Reference Artefact
#79	System architecture	Referenc Artefact
#80	System Life Cycle Process	Reference Activity
#81	System Requirements	Reference Artefact
#82	System Safety Assessment Process	Reference Activity
#83	System safety assessment process	Reference Activity
#84	Test case	Reference Activity
#85	Testing	Reference Activity
#86	Type design	Reference Activity

Tabla 2. Términos del estándar DO-178C

Cada vez que se identifica un término, se añade a la Terminología y se asocia el clúster semántico al que pertenezca y al clúster propio del estándar, tal y como se encuentra explicado en el punto 3.4.

Este proceso se repetirá sucesivamente hasta que ya no queden más términos en el standard que analizar e identificar.

A continuación, se muestra una tabla a modo de resumen de los términos registrados en la herramienta KM según su clúster:

Clúster	N.º de registros
Reference Artefact	50
Reference Activity	27
Reference Role	9
Reference Technique	6
Reference Attribute	4

Tabla 3.- Resumen términos registrados en KM; Estándar DO-178C.

2.2 Modelar relaciones entre los términos:

Después de que todos los términos del estándar DO-178C hayan sido introducidos y clasificados, el siguiente paso consiste en establecer las relaciones existentes entre ellos según el modelo conceptual (Punto 3.2).

Todas las relaciones serán implementadas en la herramienta según se ha explicado en la metodología (Punto 3.4).

-Aplicación:

Esta tarea tiene tres aspectos principales a tratar:

0. Modelado de Acrónimos: En esta tarea se establecerán las relaciones entre los acrónimos del estándar con sus respectivos términos del glosario (Punto 2.1; 2ª Fase). La relación a utilizar será "Equivalence".

   Ejemplos:

   - ARP >>> Relación Equivalence >>> Aerospace Recommended Practice.
   - ATM >>> Relación Equivalence >>> Air Traffic Management.
   - CAST >>> Relación Equivalence >>> Certificacion Authorities Software Team.

1. Modelado de Proceso: En esta parte se establecerán las relaciones de los términos relacionados con el proceso. Los términos participantes serán los Reference Activity y Reference Artefact. (Punto 2.1; 2ª Fase).

   Las relaciones a llevar a cabo serán únicamente de output, input, subactivity (PBS) y predecesor.

   Ejemplos:

   - System Safety Assessment Process >>> Subactivity (PBS) >>> System life Cycle.
   - Software Planning Process >>> Output >>> Software Development Plan.
   - Software Development Plan >>> Input >>> Software Requirements Process.
   - System Life Cycle >>> Predecesor >>> Software Requirements Process.

2. Modelado de Artefacto: En esta parte se establecerán las relaciones entre los términos relacionados con los artefactos. Los términos participantes serán los Reference Artefact.

Las relaciones a llevar a cabo serán Taxonomía, PBS y Reference Artefact RelationShip.

En el caso de la relación Reference Artefact habrá que especificar las especializaciones según el efecto que tengan dentro del estándar. Algunos ejemplos dentro de este estándar son: Develope to, Conform to, Implements to.

Ejemplos:

- High Level Requirements >>> Develope to (Reference Artefact Relationship) >>> Software Architecture.
- Low-Level Requirement >>> Implements to (Reference Artefact Relationship) >>> Source Code.
- Derived High-level >>> PBS >>> Software Requirement Data.

A continuación, se muestra una tabla a modo de resumen de las relaciones registradas en la herramienta KM según su tipo:

Relación	N.º de relaciones
Input	6
Output	8
Equivalence	26
PBS	8
Artefact RelationShip	7
Predecessor	2

Tabla 4. Relaciones registradas en KM dentro del estándar DO-178C.

4.1.2.- Análisis y validación del estándar:

En este punto se analiza y se valida lo anteriormente realizado con el objetivo de poder obtener conclusiones a cerca de la calidad del estándar DO-178C. Para ello se utiliza la herramienta RQA. Se ha dividido el proceso en las siguientes fases:

1º Fase: Conectar ontología KM con RQA a través de RQS:

El primer paso consiste en conectar la representación realizada en KM con la herramienta RQA para su posterior análisis de calidad. Para ello se utilizará la herramienta RQS; RQS es el servidor a través del cual podremos conectar ambas herramientas (KM y RQA).

A continuación, se importará la BBDD de la representación creada en KM en RQS. Para ello se tendrá que crear una conexión activa en el servidor. Para crear una

conexión activa se accederá a la opción proporcionada por el menú de RQS: Configuración de base de Datos.

Una vez en el menú de configuración de base de datos, se hará click derecho y se seleccionará añadir conexión y el tipo de archivo al que corresponde la BBDD. Se muestra a continuación:

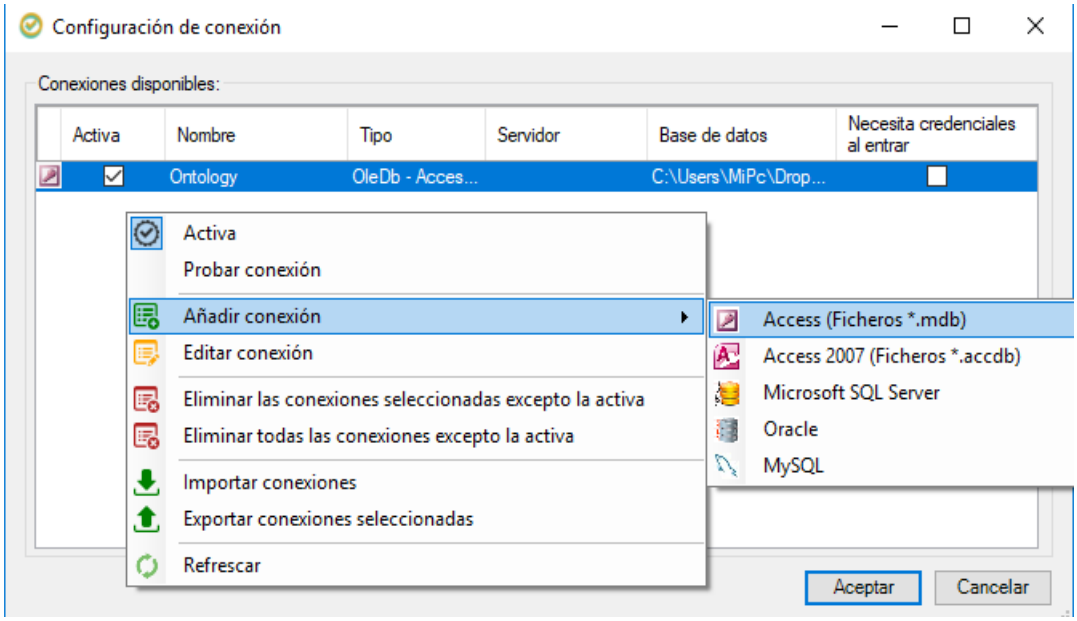


Figura 27. Configuración conexión KM con RQA a través de RQS.

Seguidamente, se le da un nombre a la conexión activa y una dirección local de dónde se encuentra la BBDD a importar al servidor.

De esta manera se tendrá la ontología creada en KM conectada a la herramienta RQA para su posterior análisis métrico de calidad.

2º Fase: Creación e importación de los requisitos del estándar DO-178C en la herramienta RQA:

El segundo paso consistirá en importar los requisitos del estándar DO-178C en la herramienta RQA. Para ello se tendrá que rellenar un archivo Excel con las siguientes características y campos de los requisitos del estándar:

Columna	Campo	Descripción
#1	ID	Identificador del requisito
#2	Custom_ID	Identificador del usurario
#3	Short text	Título del requisito
#4	Description	Requisito del Estándar
#5	Author	Nombre del usuario

#6	Creation_Date	Fecha de creación
#7	Last_Update	Fecha de modificación
#8	QualityLevel	Salidas RQA
#9	NumericQuality	Salidas RQA
#10	QualityDate	Salidas RQA
#11	QualityLevel_1	Salidas RQA
#12	NumericQuality_1	Salidas RQA
#13	QualityDate_1	Salidas RQA
#14	QualitySummary_1	Salidas RQA

Tabla 5. Características archivo Excel

Cada requisito ocupará una fila y se rellenarán los campos citados en la tabla anterior. Para más información, sobre como elaborar dicha tabla, consultar Anexo.

Una vez rellena la tabla, el siguiente paso consistirá en importarla a la herramienta RQA. Para importar el Excel en RQA crearemos un nuevo repositorio de tipo Excel. Se muestra a continuación:

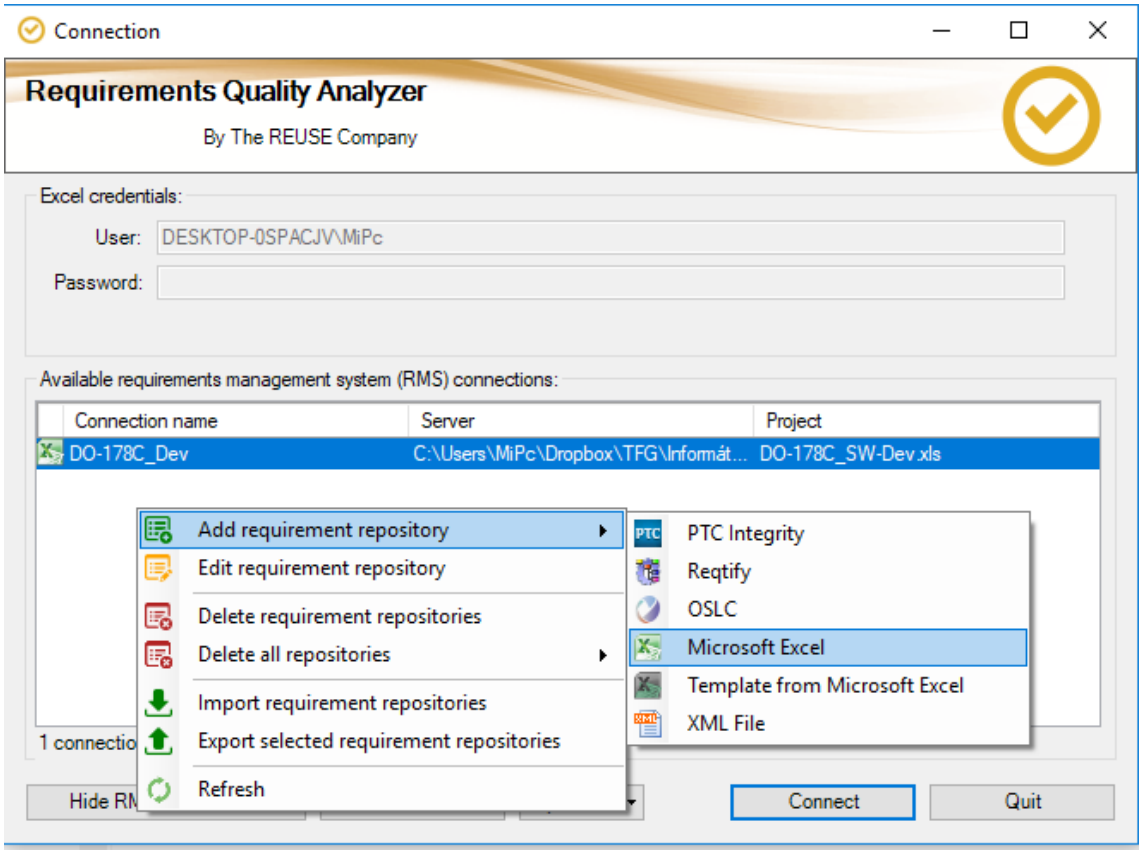


Figura 28. Importación Excel a RQA con los campos de los requisitos del estándar.

A continuación, se le da un nombre al repositorio y una dirección local de dónde se encuentra el Excel a importar.

3º Fase: Selección de métricas para el análisis:

Una vez importado el Excel en RQA, el siguiente paso consistirá en seleccionar y configurar las métricas que van a ser utilizadas en nuestro análisis de calidad.

Las métricas que se han sido seleccionadas para este análisis han sido las siguientes:

1. R01 Precision – Infinitive Articles (Avoid): Esta métrica castiga el uso del artículo indefinido "A" en lugar del artículo definido "THE", porque puede generar ambigüedad.

2. R02 Precision - Passive voice (Avoid): Esta métrica controla la existencia de voz pasiva. La voz activa requiere que el agente / actor / entidad que realiza la acción sea el sujeto de la oración, ya que los medios para satisfacer el requisito están en el sujeto, no en el objeto de la oración. Si el actor / entidad responsable del sistema no se identifica explícitamente, no está claro quién / qué debería realizar la acción.
3. R02 Precision - TRC - Conditional mode (Avoid): Esta métrica identifica requisitos que no expresan asertividad. Si un requisito es obligatorio o no, no se expresará en modo condicional.

4. R02 Precision - TRC - Imperative mode (Enforce): Esta métrica identifica los requisitos que no expresan obligación en la declaración.

5. R05 Precision - Imprecise quantifiers (Avoid): Esta métrica controla el uso de cuantificadores imprecisos.

6. R06 Precision - Units: Numbers with Measurement Units (Enforce): Esta métrica impone la asignación de unidades de medida o calificaciones de sustantivo a todos los números en una declaración de requisito.

7. R07 Precision - Vague adverbs (Avoid): Esta métrica controla la existencia de adverbios imprecisos en la declaración de requisitos. Los adverbios califican las acciones de alguna manera. Evita los adverbios vagos.

8. R08 Precision - Vague adjectives (Avoid): Esta métrica controla la existencia de adjetivos imprecisos en la declaración de requisitos. Los adjetivos califican entidades (Agentes) de alguna manera. Evita los adjetivos vagos.

9. R10 Precision - Open ended (Avoid): Esta métrica busca la existencia de cláusulas de final abierto.

10. R11 Concision - Superfluous infinitives (Avoid): A veces, un requisito tiene más verbos de lo necesario para describir una acción básica. Esta propiedad también se puede considerar una medida de singularidad.

11. R13 Non Ambiguity - TRC - Ambiguous sentences (Avoid): Esta métrica identifica requisitos ambiguos mediante el análisis de los términos de la declaración que podrían confundir el significado de la declaración.

12. R13 Non Ambiguity - TRC - Negative sentences (Avoid): Esta métrica identifica requisitos ambiguos al analizar las oraciones negativas en la misma declaración de requisitos. Tener varias expresiones negativas puede generar confusión.

13. R13 Non Ambiguity - TRC - Speculative sentences (Avoid): Esta métrica identifica requisitos ambiguos al analizar los términos especulativos en la declaración de requisitos.

14. R13 Non Ambiguity - TRC - Subjective sentences: Esta métrica identifica requisitos ambiguos al analizar los términos subjetivos en la declaración de requisitos.

15. R14 Non Ambiguity - Incorrect spelling (Avoid): La ortografía incorrecta puede generar confusión y, por lo tanto, aumenta la ambigüedad. Esta métrica busca la ortografía incorrecta en la declaración de requisitos y cuenta la cantidad de términos erróneos encontrados.

16. R15 Non Ambiguity - Incorrect Punctuation (number of characters between two punctuation symbols): Esta métrica controla el número de caracteres entre dos símbolos de puntuación dentro de la declaración de requisitos.

17. R15 Non Ambiguity - Incorrect Punctuation (Readability) (Avoid): La puntuación incorrecta puede causar confusión entre las sub-cláusulas en un requisito.

18. R16 Non Ambiguity - Conjunction "both X and Y" (Avoid): Esta métrica calcula si la expresión "tanto X como Y" se encuentra en una declaración de requisito y si se encuentra, establece su calidad como baja.

19. R17 Non Ambiguity - Statement and/or (Avoid): Esta métrica controla la existencia de la cláusula "y / o" dentro de una declaración de requisito.

20. R18 Non Ambiguity - Oblique Symbol / (Avoid): Esta métrica controla la existencia de la cláusula del símbolo oblicuo "/" dentro de una declaración de requisito.

21. R19 Singularity - TRC - Text length (paragraphs): Esta métrica identifica los requisitos que están estructurados en más de un párrafo para evitar la sobre especificación, varias necesidades en el mismo requisito o incluso información inútil.

22. R19 Singularity - TRC - Text length (words): Esta métrica identifica los requisitos con declaraciones demasiado largas al contar el número de palabras, para evitar la sobre-especificación, varias necesidades en el mismo requerimiento o incluso información inútil.

23. R20 Singularity - Combinators (Avoid): La presencia o combinators en un requisito generalmente indica que se deben escribir múltiples requisitos. Demasiados combinatos deben evitarse en un requisito.

24. R23 Singularity – Parenthesis: Si el texto de un requisito contiene corchetes, generalmente indica la presencia de información superflua que simplemente puede eliminarse o comunicarse en el fundamento.

25. R26 Completeness - Pronouns (Avoid): Esta métrica verifica la existencia de pronombres en la declaración de requisitos. Esta propiedad también se puede considerar como una medida para la no ambigüedad.

26. R33 Abstraction Level - Solution vocabulary: Cada esfuerzo del sistema debe tener un nivel de requisitos que capture el problema a resolver sin involucrar soluciones.

27. R34 Quantifiers - Ambiguous Universal Keywords (Avoid): Esta métrica verifica la existencia de palabras clave universales ambiguas en la declaración de requisitos. Esta propiedad también se puede considerar como una medida para la no ambigüedad.

28. R36 Quantification - Un-measurable terms (Avoid): Esta métrica verifica la existencia de términos no medibles en la declaración de requisitos. Esta propiedad también se puede considerar como una medida para la no ambigüedad.

29. R37 Quantification - Indefinite temporal keywords (Avoid): Esta métrica verifica la existencia de palabras clave temporales indefinidas en la declaración de requisitos. Esta propiedad también se puede considerar como una medida para la no ambigüedad.

30. R64 Concision - TRC - Rationale sentences (Avoid): Esta métrica identifica las oraciones lógicas en la declaración de requisitos que pueden llevar a la confusión, como "a fin de" o "por eso".

31. DO-178C term: Esta métrica identifica si en el requisito existe un término del estándar DO-178C.

4º Fase: Ejecución y resultados del análisis:

Después de haber seleccionado las métricas el siguiente paso consiste en la ejecución de la herramienta RQA. El análisis se ejecutará a través de la opción “Evaluar CCC para la especificación completa”, el resultado se muestra a continuación:

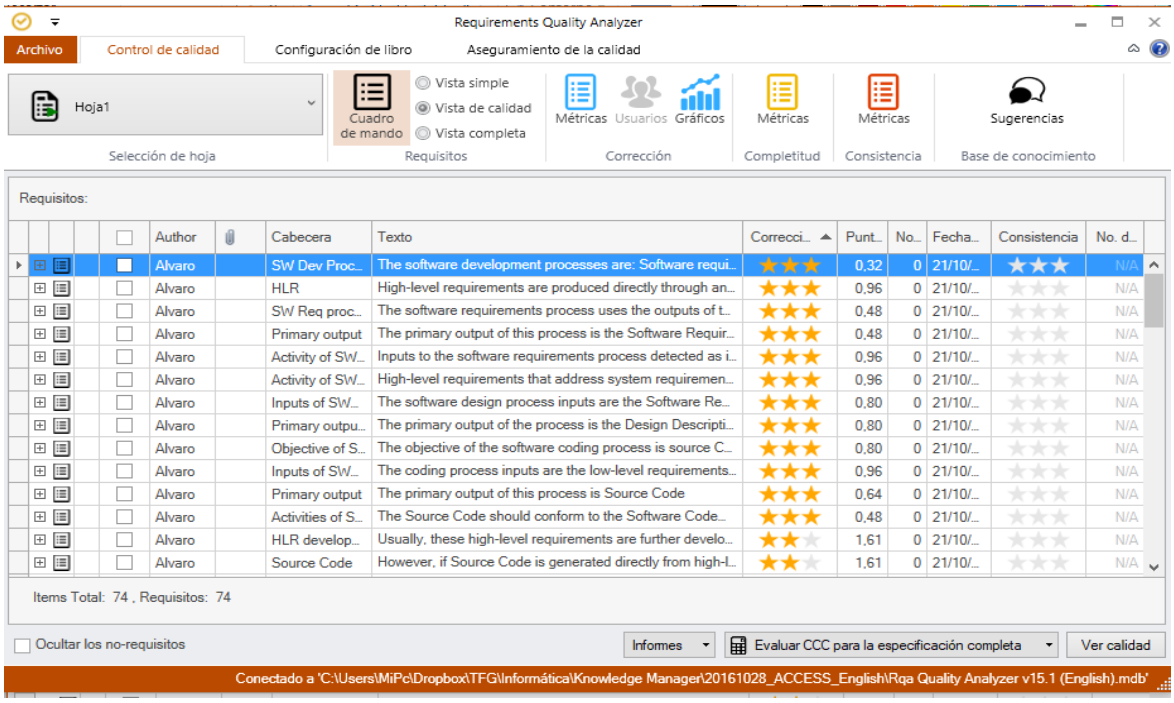


Figura 29. Resultado del análisis de RQA.

El programa nos mostrará cual ha sido la puntuación de cada requisito dentro del conjunto de métricas seleccionadas.

Para profundizar en el análisis se hará click en la opción “Métricas” y después en “Generar informe”. De esta manera tendremos un informe global sobre la calidad del estándar. Se muestra a continuación:

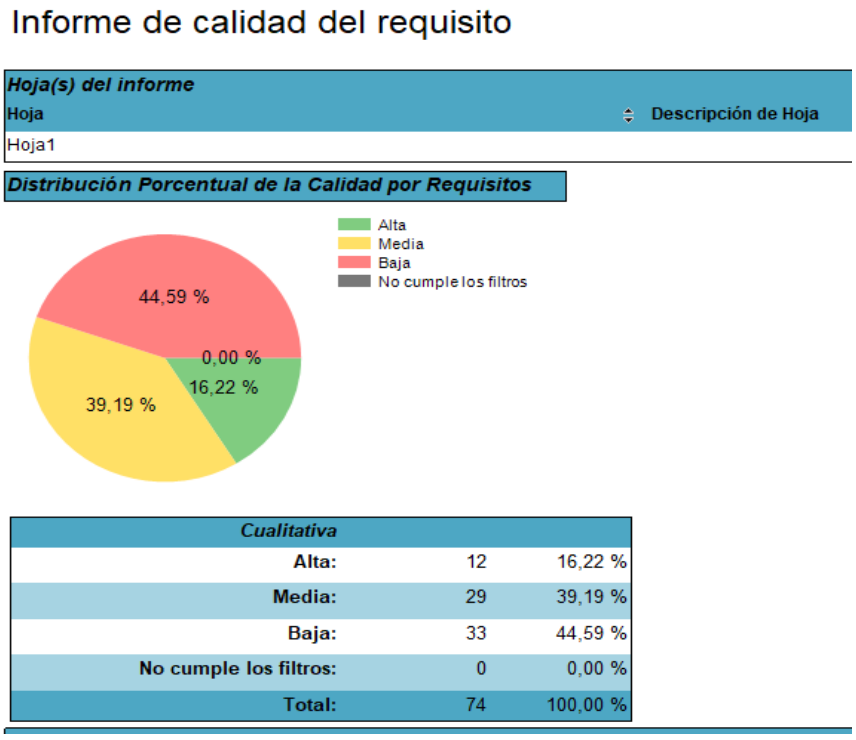
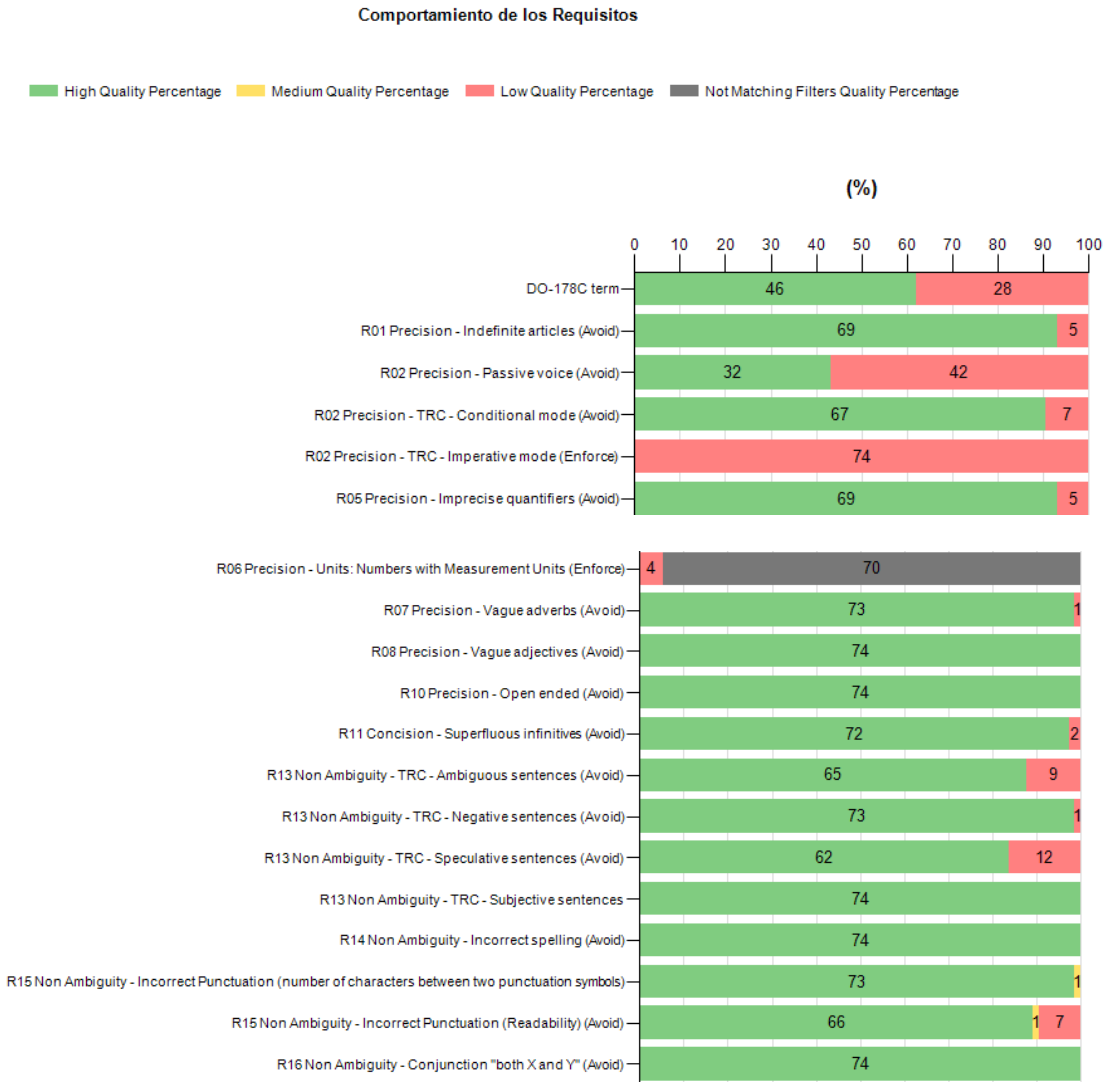


Figura 30. Informe de calidad del requisito del estándar

En el gráfico se puede observar como el **16,22%** de los requisitos tiene una calidad alta, el **39,19%** una calidad media y el **44,59%** restante calidad baja. Con estas estadísticas poder concluir que la calidad del estándar DO-178C es media.

También podemos observar el resultado obtenido métrica por métrica para poder conocer así cuales son los errores más o menos frecuentados dentro del estándar. Las estadísticas se muestran a continuación:



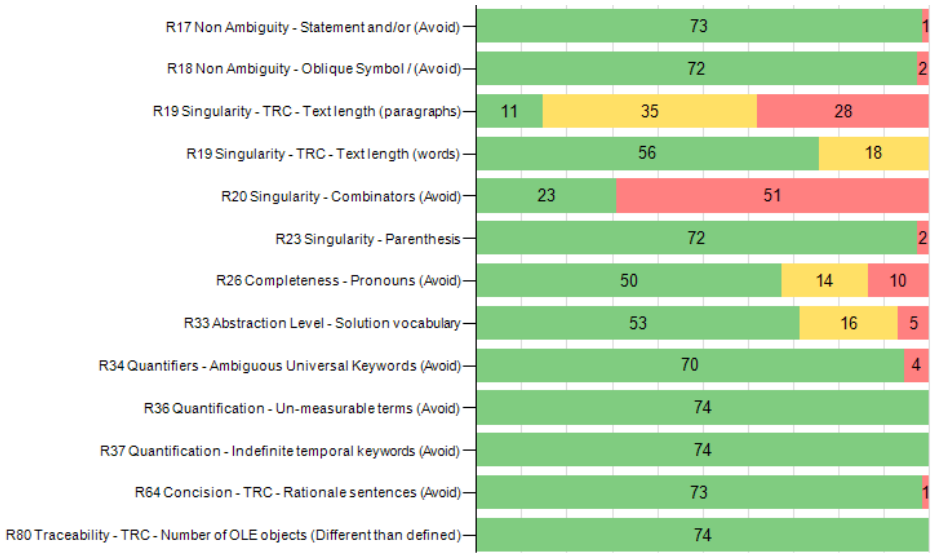


Figura 31. Comportamiento de las métricas en base a los requisitos seleccionados del estándar.

Con este informe podemos concluir que el error más frecuente se produce en la métrica **R02 Precision - TRC - Imperative mode (Enforce)**. Esta métrica se corresponde a la falta del uso del tiempo modal imperativo, esto puede crear ambigüedades puesto que es necesario expresar obligación a la hora de enunciar un requisito, ya que sino puede quedar en duda su cumplimiento.

Otros errores, menos frecuentes que el anterior, que podemos observar se encuentran en la métrica **R02 Precision – Passive** Voice y **R20 Singularity – Combinators**.

La primera métrica se corresponde con evitar el uso de la forma verbal pasiva. El uso de la forma pasiva puede crear ambigüedad puesto que esta forma verbal no se deja connotado el sujeto de la acción por lo que queda en duda sobre quien va llevar a cabo ese requisito. Podemos observar como en el 40% de los casos no aparece, pero en el 60% restante sí.

La segunda métrica corresponde con el uso de expresiones de combinación. Las expresiones de combinación hay que evitarlas puesto que pueden hacer el requisito demasiado largo y complejo, pudiendo crear ambigüedad para la persona que lo está aplicando. En este caso podemos observar cómo el 30% de los casos no aparecen, pero en el 70% restante sí.

Los errores menos frecuentes aparecen en las métricas:

- **R08 Precision - Vague adjectives (Avoid):** Esta métrica evita los adjetivos vagos que pueden causar ambigüedad. Algunos ejemplos de adjetivos vagos son los siguientes: relevante, común, genérico, habitual…

- **R10 Precision - Open ended (Avoid):** Esta métrica se cerciora de que no existan clausulas abiertas que puedan conducir al error. Ejemplo de cláusulas abiertas serían etc., sucesivamente…
- **R13 Non-Ambiguity - TRC - Subjective sentences:** Esta métrica evita el uso de oraciones subjetivas en las que el autor pueda dar su opinión o generar duda.
- **R14 Non Ambiguity - Incorrect spelling (Avoid):** Esta métrica corrige los errores ortográficos. Los errores ortográficos pueden causar confusión.
- **R36 Quantification - Un-measurable terms (Avoid):** Esta métrica verifica la existencia de términos no medibles en la declaración de requisitos.
- **R37 Quantification - Indefinite temporal keywords (Avoid):** Esta métrica verifica la existencia de palabras clave temporales indefinidas en la declaración de requisitos. Ejemplo de palabras claves temporales son: a veces, siempre…
- **R80 Traceability - TRC - Number of OLE objects (Different than defined):** Esta métrica identifica el número de objetos OLE y analiza si los números incluidos en el requisito son diferentes a los definidos en la calidad.

4.2.- Caso 2: EN-50128:

En este apartado se realiza lo anteriormente explicado en una subsección del Estándar EN 50128. La razón por la que ha sido seleccionada esta subsección ha sido porque existen modelos disponibles de este estándar validado por profesionales.

4.2.1. Representación del estándar:

En este apartado se lleva a cabo la representación ontológica del estándar EN-50128. Para ello se utiliza la metodología explicada en el punto 3 de este TFG.

La metodología explicada consta de dos fases a través de las cuales se realiza la representación:

1ª Fase: Configuración KM:

En esta primera fase se realizará la configuración de la herramienta KM en base al estándar analizado. Se deben realizar dos tareas:

1.1 Especificación de grupos semánticos:

En esta tarea los clústeres del estándar EN-50128 deben agregarse a la capa del modelo conceptual para poder indicar el tipo de información que representa un término.

En primer lugar, es necesario un clúster con el nombre del estándar de seguridad que se va a representar para especificar más adelante que un término cae dentro del alcance de la norma.

En segundo lugar, se introducirán los tipos de clúster pertenecientes al metamodelo. Estos clústeres tendrán la finalidad de modelar los diferentes elementos de una RAF.

-Aplicación:

1. El primer paso consiste en crear un clúster con el nombre del estándar, para ello seguiremos los pasos establecidos por la metodología para crear un clúster, punto 3.4.

   El resultado final sería el siguiente que se ve a continuación:

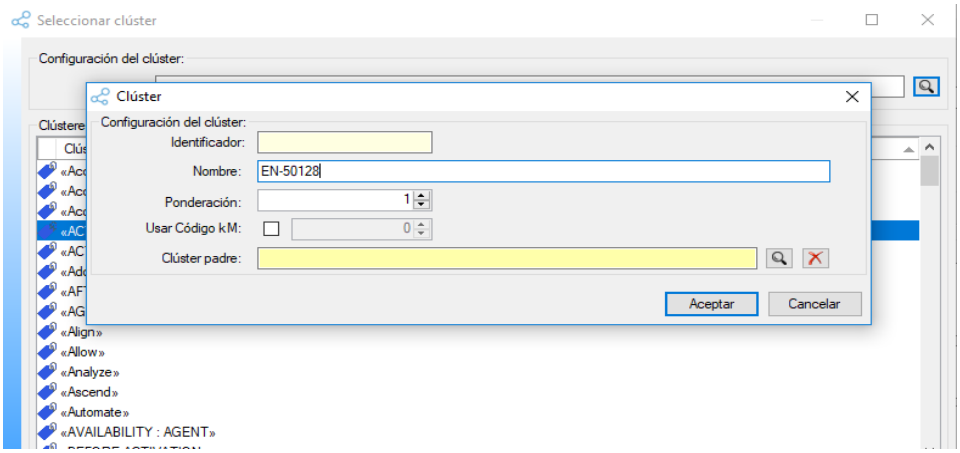


Figura 32. Creación del clúster propio del estándar.

2. Una vez creado el clúster del estándar se pasará a crear los clústeres del metamodelo holístico presentado en el punto 3.2.

   En el caso del estándar EN-50128 los clústeres que aparecen son los siguientes:

   - Reference Artefact: El clúster "Reference Artefact" hace referencia a las unidades de datos que podrían tener que ser administradas dentro del estándar EN-50128.

     En este caso el clúster Reference Artefact se ha de configurar en la herramienta KM puesto que aparecen muchos términos que estarían asociados a este.

     Para crear el clúster se seguirán los pasos estipulados por la metodología explicada en el punto 3.4.

Algunos ejemplos dentro del estándar que se corresponderían con este clúster serían: Hardware Components, Integrated System, Software/Hardware Integration Test Report….

- Reference Activity: El clúster "Reference Activity" hace referencia a las unidades de comportamiento que podría tener que ser ejecutadas dentro del estándar EN-50128.

  En este caso el clúster Reference Activity se ha de configurar en la herramienta KM puesto que aparecen muchos términos que estarían asociados a este.

  Para crear el clúster se seguirán los pasos estipulados por la metodología explicada en el punto 3.4.

  Algunos ejemplos dentro del estándar que se corresponderían con este clúster serían: Integration Process, Software Integration Verification, Software Integration Testing…

- Reference Role: El clúster "Reference Role" hace referencia a los tipos de agentes que podrían tener que estar involucrados en los procesos dentro del estándar EN-50128.

  En este caso el clúster Reference Role se ha de configurar en la herramienta KM puesto que aparecen términos que estarían asociados a este.

  Para crear el clúster se seguirán los pasos estipulados por la metodología explicada en el punto 3.4.

  Algunos ejemplos dentro del estándar que se corresponderían con este clúster serían: Integrator y Verifier.

- Reference Technique: El clúster "Reference Technique" hace referencia a las formas específicas de ejecutar una actividad de referencia o crear un artefacto de referencia dentro del estándar EN-50128.

  En este caso el clúster Reference Technique se ha de configurar en la herramienta KM puesto que aparecen términos que estarían asociados a este.

  Para crear el clúster se seguirán los pasos estipulados por la metodología explicada en el punto 3.4.

Algunos ejemplos dentro del estándar que se corresponderían con este clúster serían: Performance Testing y Funcional and Black-Box Testing.

El resultado final sería el siguiente que se ve a continuación:

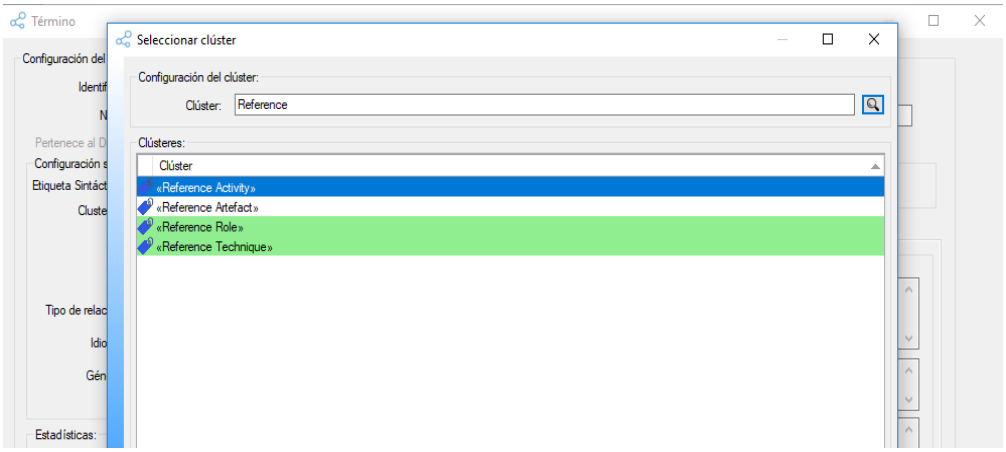


Figura 33. Creación de los clústeres del metamodelo pertenecientes al estándar.

1.2 Especificación de tipos de relaciones:

En esta segunda tarea se introducirán los tipos de relaciones pertenecientes al metamodelo (Punto 3.2) entre los diferentes tipos de clústeres. Dichas relaciones tendrán la finalidad de asociar los diferentes elementos del estándar EN 50128.

-Aplicación:

Las relaciones que configurar dentro de este estándar son las siguientes:

- Taxonomía (Ya definida por la herramienta): Se define esta relación para representar las posibles taxonomías entre términos del estándar. Esta relación se lleva a cabo entre términos pertenecientes al clúster Reference Artefact.
- PBS (Ya definida por la herramienta): Se define esta relación para representar la composición entre término. Esta relación se lleva a cabo entre términos pertenecientes al clúster Reference Artefact.
- Input: Se define esta relación para representar la entrada de las actividades propias del estándar. Esta relación se lleva a cabo entre términos pertenecientes al clúster Reference Artefact - Reference Activity.
- Output: Se define esta relación para representar la salida de las actividades propias del estándar. Esta relación se lleva a cabo entre términos pertenecientes al clúster Reference Artefact - Reference Activity.

- Predecesor: Se define esta relación para representar el orden las actividades propias del estándar. Esta relación se lleva a cabo entre términos pertenecientes al clúster Reference Activity.
- Participant: Se define esta relación para representar la relación existente entre un rol y una actividad propia del estándar. Esta relación se lleva a cabo entre términos pertenecientes a los clústeres Reference Role y Reference Activity.
- Technique: Se define esta relación para representar la relación existente entre una técnica y una actividad propia del estándar. Esta relación se lleva a cabo entre términos pertenecientes a los clústeres Reference Technique y Reference Activity.
- Reference Artefact RelationShip: Se define esta relación para representar la relación existente entre los términos pertenecientes al clúster Reference Artefact.

Dichas relaciones serán configuradas en la herramienta KM según la metodología explicada en el punto 3.4.

El resultado final sería el siguiente que se ve a continuación:

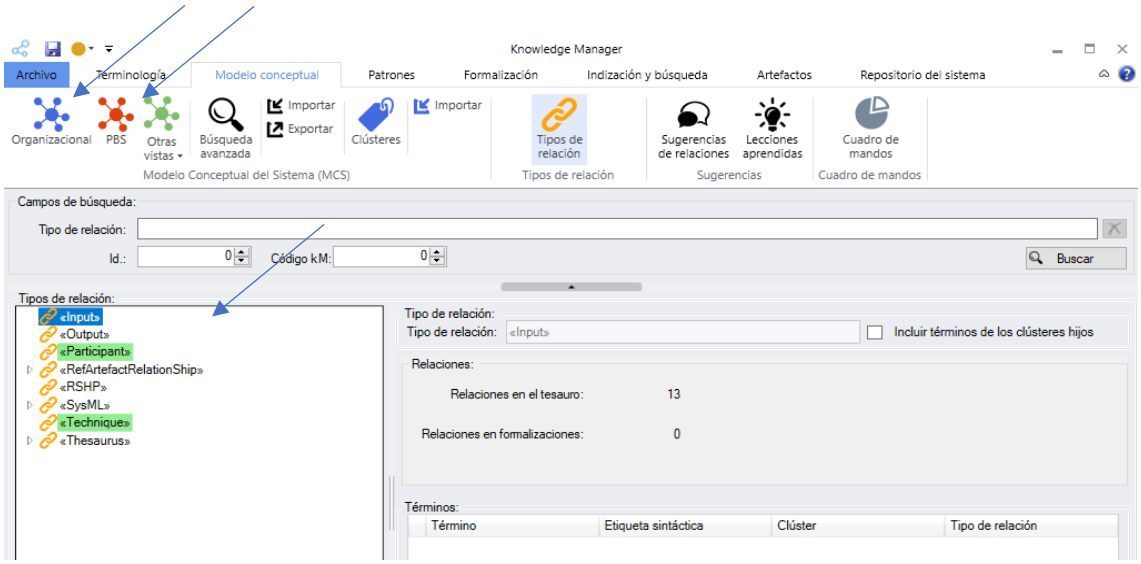


Figura 34. Configuración de las relaciones del metamodelo pertenecientes al estándar.

2ª Fase: Especificación de la información de una norma:

Una vez configurada la herramienta KM en base al estándar EN 50128, el siguiente paso consiste en la especificación de la información en base al metamodelo del punto 3.2, para ser después modelada en la herramienta.

En esta fase se distinguen dos tareas. Normalmente, las tareas se ejecutarán iterativamente para representar incrementalmente el estándar de seguridad.

2.1 Especificación de la terminología de un estándar:

En esta primera actividad se analizará el texto en busca de términos específicos del estándar EN 50128. Una vez identificados serán modelados en un clúster u en otro dependiendo de cuál sea su finalidad dentro del estándar.

Un término pertenecerá a un clúster o a otro dependiendo de sus características dentro del contexto del estándar. Las características vienen dadas en el punto 3.4 de la metodología.

-Aplicación:

Esta tarea tiene dos aspectos principales a tratar:

1. Términos glosario: En primer lugar, se analizarán y se clasificarán los términos del glosario con sus respectivos acrónimos, en su clúster correspondiente.

   El glosario dentro del estándar EN 50128 se encuentra la página 11: "Abbreviations".

   Dentro el estándar EN 50128 los términos del glosario encontrados con sus respectivos clústeres han sido los siguientes:

Acrónimo	Significado	Clúster
ASR	Assesor	Reference Role
COTS	Commercial off-the-shelf	-
DES	Designer	Reference Role
HR	Highly Recommended	-
IMP	Implementer	Reference Role
INT	Integrator	Reference Role
JSD	Jackson System Development Method	Reference Technique
MASCOT	Modular Approach to Software Construction, Operation and Test	Reference Technique
PM	Proyect Manager	Reference Role
RAMS	Reliability, Availability, Maintainability and Safety	-
RQM	Requirements Manager	Reference Role
SDL	Specification and Description Language	Reference Technique
SFC	Sequential Function Charts	Reference Technique
SOM	Service Oriented Modeling	Reference Technique
SSADM	Structured Systems Analysis & Design Methodology	Reference Technique
TST	Tester	Reference Role
V&V	Verification and Validation	Reference Activity

VAL	Validator	Reference Role
VER	Verifier	Reference Role

Tabla 6. Acrónimos del estándar EN-50128

Una vez identificados y clasificados los términos del glosario, el siguiente paso consiste en registrar tanto el acrónimo como el término en la herramienta KM con su clúster correspondiente, para ello seguiremos los pasos estipulados en el punto 3.4 de la metodología.

2. Términos del estándar: En segundo lugar, se analizará el texto del estándar EN 50128 con el objetivo de identificar los términos a representar en la aproximación.

Dentro el estándar EN 50128 los términos encontrados con sus respectivos clústeres han sido los siguientes:

Número	Término	Clúster
#1	Assembled Software	Reference Artefact
#2	Change report	Reference Artefact
#3	Configuration manager	Reference Role
#4	Customer	Reference Role
#5	Entity	Reference Role
#6	Firmware	Reference Artefact
#7	Functional and Black-Box Testing	Reference Technique
#8	Generic Software	Reference Artefact
#9	Hardware Components	Reference Artefact
#10	Implementer	Reference Role
#11	Integrated System	Reference Artefact
#12	Integration Process	Reference Activity
#13	Open Source Software	Reference Artefact
#14	Performance Testing	Reference Technique
#15	Programmable Logic Controller	Reference Artefact
#16	Project Management	Reference Activity
#17	Reliability	Reference Attributte
#18	Requirements Management	Reference Activity
#19	Requirements Manager	Reference Role
#20	Robustness	-
#21	Safety Authority	Reference Role
#22	Software and Hardware Integration	Reference Activity
#23	Software and Hardware Integration Test Report	Reference Artefact
#24	Software and Hardware Integration Test Specification	Reference Artefact
#25	Software and Hardware Integration Testing	Reference Activity

#26	Software Baseline	Reference Artefact
#27	Software Components	Reference Artefact
#28	Software Deployment	Reference Artefact
#29	Software Integration	Reference Activity
#30	Software Integration Test Report	Reference Artefact
#31	Software Integration Test Specification	Reference Artefact
#32	Software Integration Testing	Reference Activity
#33	Software Integration Verification	Reference Activity
#34	Software Integration Verification Report	Reference Artefact
#35	Software Life-cycle	Reference Activity
#36	System Safety Integrity Level	Reference Attributte

Tabla 7. Términos del estándar EN-50128

Cada vez que se identifica un término, se añade a la Terminología y se asocia el clúster semántico al que pertenezca y al clúster propio del estándar, tal y como se encuentra explicado en el punto 3.4.

Este proceso se repetirá sucesivamente hasta que ya no queden más términos en el standard que analizar e identificar.

A continuación, se muestra una tabla a modo de resumen de los términos registrados en la herramienta KM según su clúster:

Clúster	N.º de registros
Reference Artefact	16
Reference Activity	10
Reference Role	16
Reference Technique	8
Reference Attribute	2

Tabla 8. Resumen términos registrados en KM; Estándar EN-50128.

2.2 Modelar relaciones entre los términos:

Después de que todos los términos del estándar EN 50128 hayan sido introducidos y clasificados, el siguiente paso consiste en establecer las relaciones existentes entre ellos según el modelo conceptual (Punto 3.2).

Todas las relaciones serán implementadas en la herramienta según se ha explicado en la metodología (Punto 3.4).

-Aplicación:

Esta tarea tiene tres aspectos principales a tratar:

1. Modelado de Acrónimos: En esta tarea se establecerán las relaciones entre los acrónimos del estándar con sus respectivos términos del glosario (Punto 2.1; 2ª Fase). La relación a utilizar será "Equivalence".

   Ejemplos:

   - ASR >>> Relación Equivalence >>> Assesor.
   - DES >>> Relación Equivalence >>> Designer.
   - IMP >>> Relación Equivalence >>> Implementer.

2. Modelado de Proceso: En esta parte se establecerán las relaciones de los términos relacionados con el proceso. Los términos participantes serán los Reference Activity, Reference Artefact, Reference Role y Reference Technique (Punto 2.1; 2ª Fase).

   Las relaciones a llevar a cabo serán únicamente de "Output", "Input", "Subactivity" (PBS), "Predecesor", "Participant" y "Technique".

   Ejemplos:

   - Integration Process >>> Subactivity (PBS) >>> Software Integration.
   - Software Integration Verification >>> Output >>> Software Integration Verification Report.
   - Software Components >>> Input >>> Software Integration.
   - Verifier >>> Participant >>> Integration Process.
   - Performance Testing >>> Technique >>> Integration Process.

3. Modelado de Artefacto: En esta parte se establecerán las relaciones entre los términos relacionados con los artefactos. Los términos participantes serán los Reference Artefact.

   Las relaciones a llevar a cabo serán Taxonomía, PBS y Reference Artefact RelationShip.

   En el caso de la relación Reference Artefact habrá que especificar las especializaciones según el efecto que tengan dentro del estándar. Algunos ejemplos dentro de este estándar son: "ShowTestingResultsFor", "Regarding", "Verifies", "Based on".

Ejemplos:

- Assembled Software >>> ShowTestingResultsFor (Reference Artefact RelationShip) >>> Software Integration Test Report.
- Software Integration Verification Report >>> Based on (Reference Artefact RelationShip) >>> Software Integration Test Specification.
- Software/Hardware Integration Test Report >>> Verifies >>> Software Integration Verification Report.
- Hardware Components >>> Includes (PBS) >>> Integrated System.

A continuación, se muestra una tabla a modo de resumen de las relaciones registradas en la herramienta KM según su tipo:

Relación	N.º de relaciones
Input	11
Output	6
Equivalence	20
PBS	5
Artefact RelationShip	12
Participant	4
Technique	2

Tabla 9. Relaciones registradas en KM dentro del estándar EN-50128.

4.2.2.- Análisis y validación del estándar:

En este punto se analiza y se valida lo anteriormente realizado con el objetivo de poder obtener conclusiones a cerca de la calidad del estándar EN 50128. Para ello se utiliza la herramienta RQA. Se ha dividido el proceso en las siguientes fases:

1º Fase: Conectar ontología KM con RQA a través de RQS:

El primer paso consiste en conectar la representación realizada en KM con la herramienta RQA para su posterior análisis de calidad. Para ello se utilizará la herramienta RQS; RQS es el servidor a través del cual podremos conectar ambas herramientas (KM y RQA).

A continuación, se importará la BBDD de la representación creada en KM en RQS, para ello se tendrá que crear una conexión activa en el servidor. Para crear una conexión activa se accederá a la opción proporcionada por el menú de RQS: Configuración de base de Datos.

Una vez en el menú de configuración de base de datos, se hará click derecho y se seleccionará añadir conexión y el tipo de archivo al que corresponde la BBDD. Se muestra a continuación:

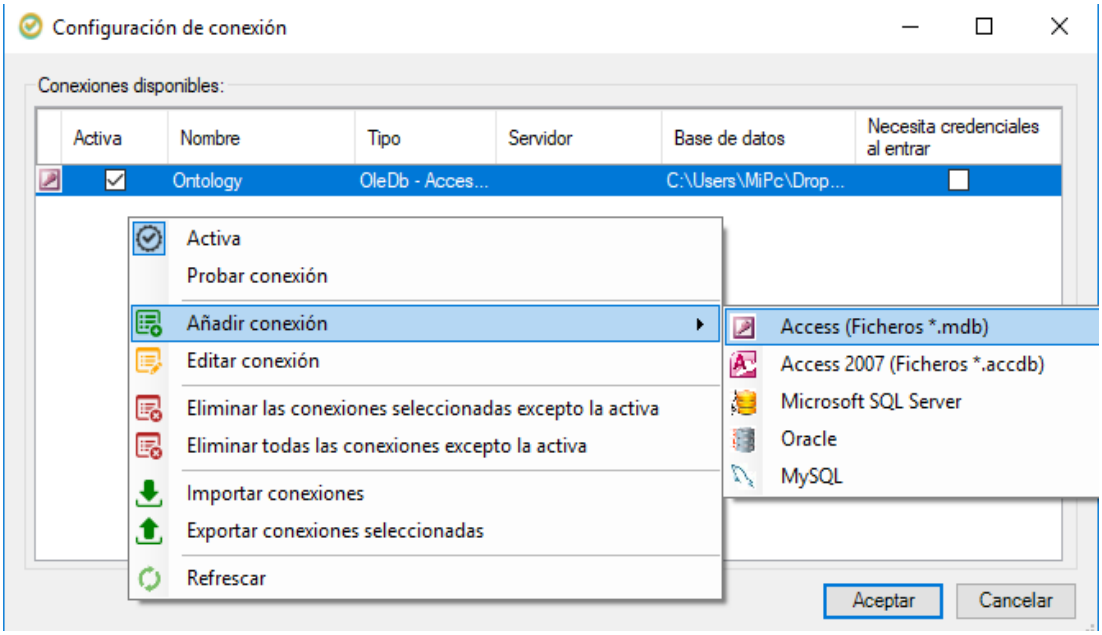


Figura 35. Configuración conexión KM con RQA a través de RQS.

Seguidamente, se le da un nombre a la conexión activa y una dirección local de dónde se encuentra la BBDD a importar al servidor.

De esta manera se tendrá la ontología creada en KM conectada a la herramienta RQA para su posterior análisis métrico de calidad.

2º Fase: Creación e importación de los requisitos del estándar EN 50128 en la herramienta RQA:

El segundo paso consistirá en importar los requisitos del estándar EN 50128 en la herramienta RQA. Para ello se tendrá que rellenar un archivo Excel con las siguientes características y campos de los requisitos del estándar:

Columna	Campo	Descripción
#1	ID	Identificador del requisito
#2	Custom_ID	Identificador del usurario
#3	Short text	Título del requisito
#4	Description	Requisito del Estándar
#5	Author	Nombre del usuario
#6	Creation_Date	Fecha de creación
#7	Last_Update	Fecha de modificación
#8	QualityLevel	Salidas RQA

#9	NumericQuality	Salidas RQA
#10	QualityDate	Salidas RQA
#11	QualityLevel_1	Salidas RQA
#12	NumericQuality_1	Salidas RQA
#13	QualityDate_1	Salidas RQA
#14	QualitySummary_1	Salidas RQA

Tabla 10. Características archivo Excel

Cada requisito ocupará una fila y se rellenarán los campos citados en la tabla anterior. Para más información, sobre como elaborar dicha tabla, consultar Anexo.

Una vez rellena la tabla, el siguiente paso consistirá en importarla a la herramienta RQA. Para importar el Excel en RQA crearemos un nuevo repositorio de tipo Excel. Se muestra a continuación:

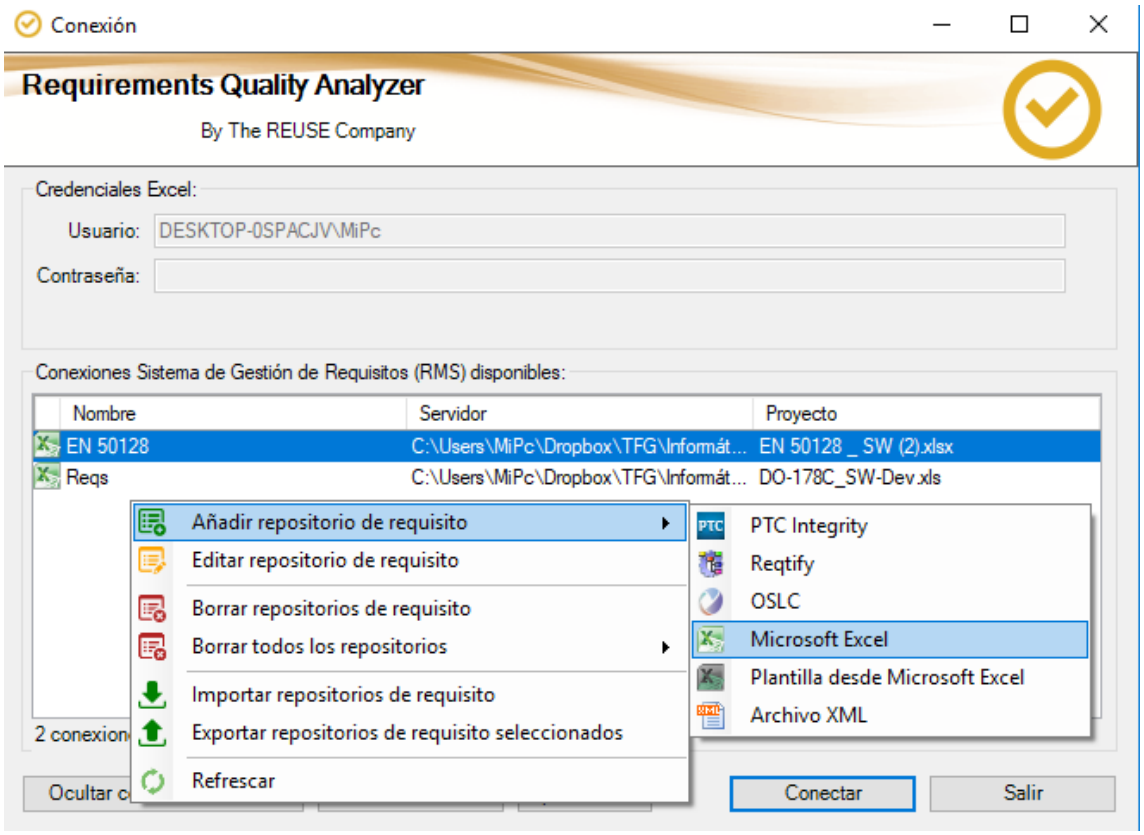


Figura 36. Importación Excel a RQA con los campos de los requisitos del estándar.

A continuación, se le da un nombre al repositorio y una dirección local de dónde se encuentra el Excel a importar.

3º Fase: Selección de métricas para el análisis:

Una vez importado el Excel en RQA, el siguiente paso consistirá en seleccionar y configurar las métricas que van a ser utilizadas en nuestro análisis de calidad.

Las métricas que se han sido seleccionadas para este análisis han sido las siguientes:

1. R01 Precision – Infinitive Articles (Avoid): Esta métrica castiga el uso del artículo indefinido "A" en lugar del artículo definido "THE", porque puede generar ambigüedad.

2. R02 Precision - Passive voice (Avoid): Esta métrica controla la existencia de voz pasiva. La voz activa requiere que el agente / actor / entidad que realiza la acción sea el sujeto de la oración, ya que los medios para satisfacer el requisito están en el sujeto, no en el objeto de la oración. Si el actor / entidad responsable del sistema no se identifica explícitamente, no está claro quién / qué debería realizar la acción.

3. R02 Precision - TRC - Conditional mode (Avoid): Esta métrica identifica requisitos que no expresan asertividad. Si un requisito es obligatorio o no, no se expresará en modo condicional.

4. R02 Precision - TRC - Imperative mode (Enforce): Esta métrica identifica los requisitos que no expresan obligación en la declaración.

5. R05 Precision - Imprecise quantifiers (Avoid): Esta métrica controla el uso de cuantificadores imprecisos.

6. R06 Precision - Units: Numbers with Measurement Units (Enforce): Esta métrica impone la asignación de unidades de medida o calificaciones de sustantivo a todos los números en una declaración de requisito.

7. R07 Precision - Vague adverbs (Avoid): Esta métrica controla la existencia de adverbios vagos en la declaración de requisitos. Los adverbios califican las acciones de alguna manera. Evita los adverbios vagos.

8. R08 Precision - Vague adjectives (Avoid): Esta métrica controla la existencia de adjetivos vagos en la declaración de requisitos. Los adjetivos califican entidades (Agentes) de alguna manera. Evita los adjetivos vagos.

9. R10 Precision - Open ended (Avoid): Esta métrica busca la existencia de cláusulas de final abierto.

10. R11 Concision - Superfluous infinitives (Avoid): A veces, un requisito tiene más verbos de lo necesario para describir una acción básica. Esta propiedad también se puede considerar una medida de Singularity.

11. R13 Non Ambiguity - TRC - Ambiguous sentences (Avoid): Esta métrica identifica requisitos ambiguos mediante el análisis de los términos de la declaración que podrían confundir el significado de la declaración.

12. R13 Non Ambiguity - TRC - Negative sentences (Avoid): Esta métrica identifica requisitos ambiguos al analizar las oraciones negativas en la misma declaración de requisitos. Tener varias expresiones negativas puede generar confusión.

13. R13 Non Ambiguity - TRC - Speculative sentences (Avoid): Esta métrica identifica requisitos ambiguos al analizar los términos especulativos en la declaración de requisitos.

14. R13 Non Ambiguity - TRC - Subjective sentences: Esta métrica identifica requisitos ambiguos al analizar los términos subjetivos en la declaración de requisitos.

15. R14 Non Ambiguity - Incorrect spelling (Avoid): La ortografía incorrecta puede generar confusión y, por lo tanto, aumenta la ambigüedad. Esta métrica busca la ortografía incorrecta en la declaración de requisitos y cuenta la cantidad de términos erróneos encontrados.

16. R15 Non Ambiguity - Incorrect Punctuation (number of characters between two punctuation symbols): Esta métrica controla el número de caracteres entre dos símbolos de puntuación dentro de la declaración de requisitos.

17. R15 Non Ambiguity - Incorrect Punctuation (Readability) (Avoid): La puntuación incorrecta puede causar confusión entre las sub-cláusulas en un requisito.

18. R16 Non Ambiguity - Conjunction "both X and Y" (Avoid): Esta métrica calcula si la expresión "tanto X como Y" se encuentra en una declaración de requisito y si se encuentra, establece su calidad como baja.

19. R17 Non Ambiguity - Statement and/or (Avoid): Esta métrica controla la existencia de la cláusula "y / o" dentro de una declaración de requisito.

20. R18 Non Ambiguity - Oblique Symbol / (Avoid): Esta métrica controla la existencia de la cláusula del símbolo oblicuo "/" dentro de una declaración de requisito.

21. R19 Singularity - TRC - Text length (paragraphs): Esta métrica identifica los requisitos que están estructurados en más de un párrafo para evitar la sobre especificación, varias necesidades en el mismo requisito o incluso información inútil.
22. R19 Singularity - TRC - Text length (words): Esta métrica identifica los requisitos con declaraciones demasiado largas al contar el número de palabras, para evitar la sobre-especificación, varias necesidades en el mismo requerimiento o incluso información inútil.

23. R20 Singularity - Combinators (Avoid): La presencia o combinators en un requisito generalmente indica que se deben escribir múltiples requisitos. Demasiados combinatos deben evitarse en un requisito.

24. R23 Singularity – Parenthesis: Si el texto de un requisito contiene corchetes, generalmente indica la presencia de información superflua que simplemente puede eliminarse o comunicarse en el fundamento.

25. R26 Completeness - Pronouns (Avoid): Esta métrica verifica la existencia de pronombres en la declaración de requisitos. Esta propiedad también se puede considerar como una medida para la no ambigüedad.

26. R33 Abstraction Level - Solution vocabulary: Cada esfuerzo del sistema debe tener un nivel de requisitos que capturen el problema a resolver sin involucrar soluciones.

27. R34 Quantifiers - Ambiguous Universal Keywords (Avoid): Esta métrica verifica la existencia de palabras clave universales ambiguas en la declaración de requisitos. Esta propiedad también se puede considerar como una medida para la no ambigüedad.

28. R36 Quantification - Un-measurable terms (Avoid): Esta métrica verifica la existencia de términos no medibles en la declaración de requisitos. Esta propiedad también se puede considerar como una medida para la no ambigüedad.

29. R37 Quantification - Indefinite temporal keywords (Avoid): Esta métrica verifica la existencia de palabras clave temporales indefinidas en la declaración de requisitos. Esta propiedad también se puede considerar como una medida para la no ambigüedad.

30. R64 Concision - TRC - Rationale sentences (Avoid): Esta métrica identifica las oraciones lógicas en la declaración de requisitos que pueden llevar a la confusión, como "a fin de" o "por eso".

31. R80 Traceability - TRC - Number of OLE objects (Different than defined): Esta métrica identifica el número de objetos OLE y analiza si los números incluidos en el requisito son diferentes a los definidos en la calidad.

32. EN 50128 term: Esta métrica identifica si en el requisito existe un término del estándar EN 50128.

4º Fase: Ejecución y resultados del análisis:

Después de haber seleccionado las métricas el siguiente paso consiste en la ejecución de la herramienta RQA. El análisis se ejecutará a través de la opción “Evaluar CCC para la especificación completa”, el resultado se muestra a continuación:

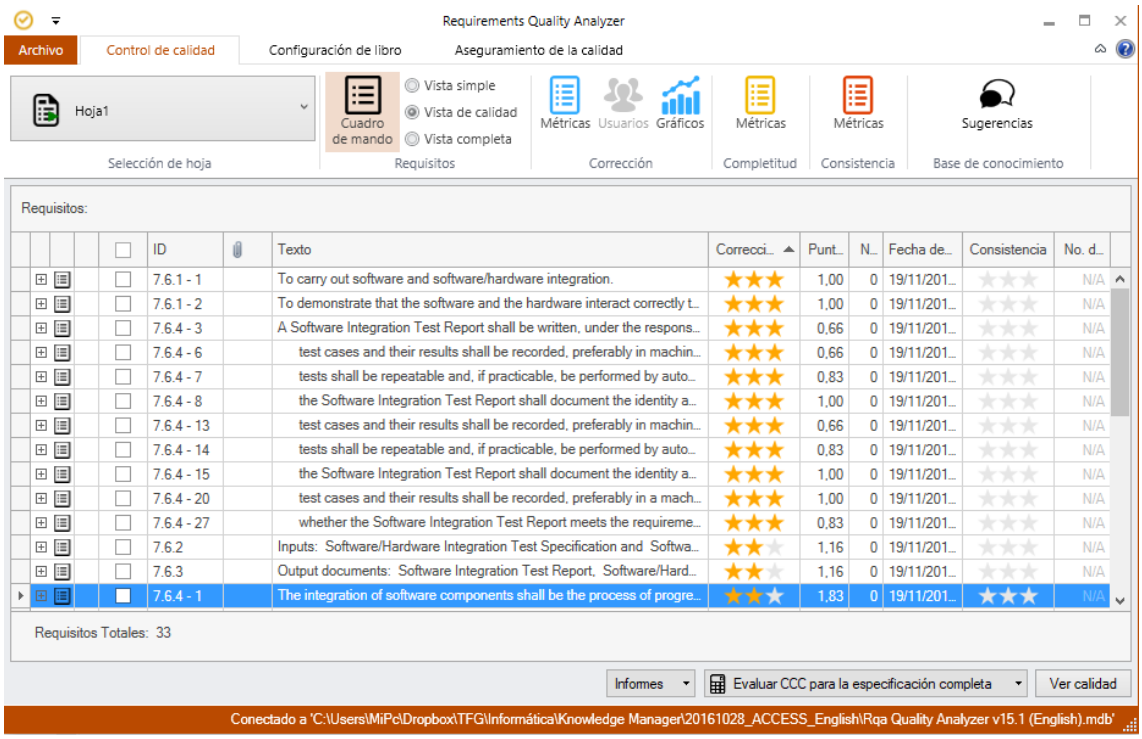


Figura 37. Resultado del análisis de RQA.

El programa nos mostrará cual ha sido la puntuación de cada requisito dentro del conjunto de métricas seleccionadas.

Para profundizar en el análisis se hará click en la opción “Métricas” y después en “Generar informe”. De esta manera tendremos un informe global sobre la calidad del estándar. Se muestra a continuación:

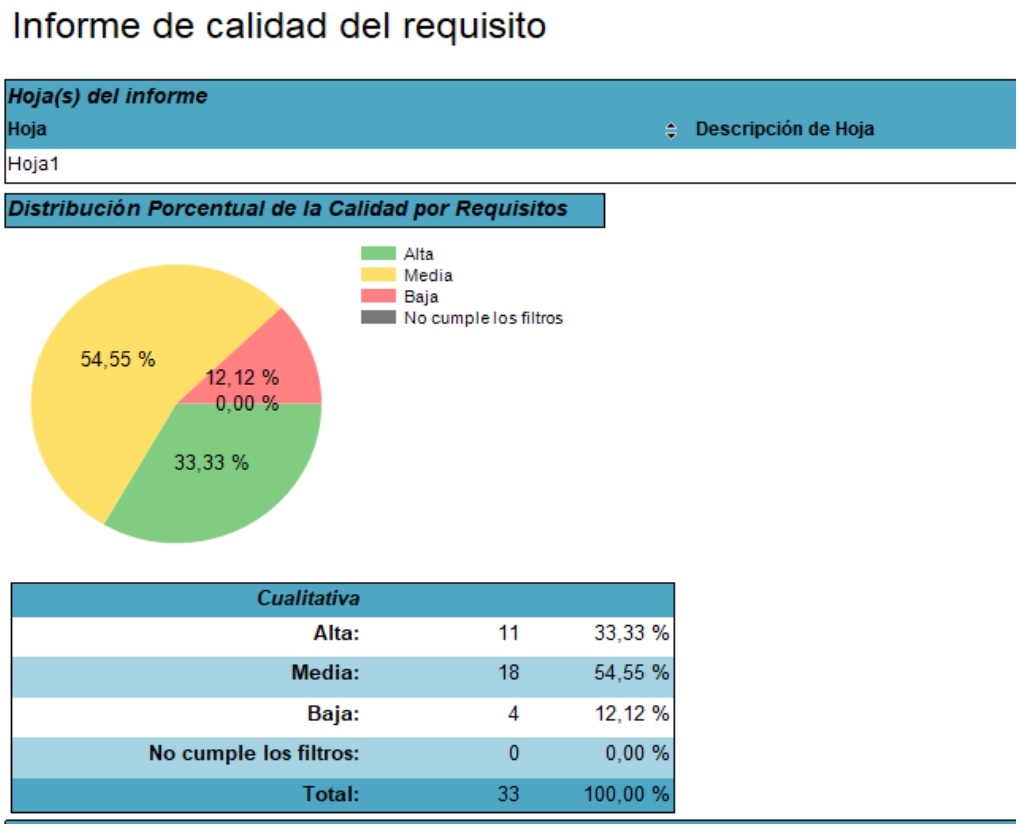


Figura 38. Informe de calidad del requisito del estándar.

En el gráfico se puede observar cómo el **33.33%** de los requisitos tiene una calidad alta, el **54.55%** una calidad media y el **12.12%** restante calidad baja. Con estas estadísticas poder concluir que la calidad del estándar EN 50128 es alta-media.

También podemos observar el resultado obtenido métrica por métrica para poder conocer así cuales son los errores más o menos frecuentados dentro del estándar. Las estadísticas se muestran a continuación:

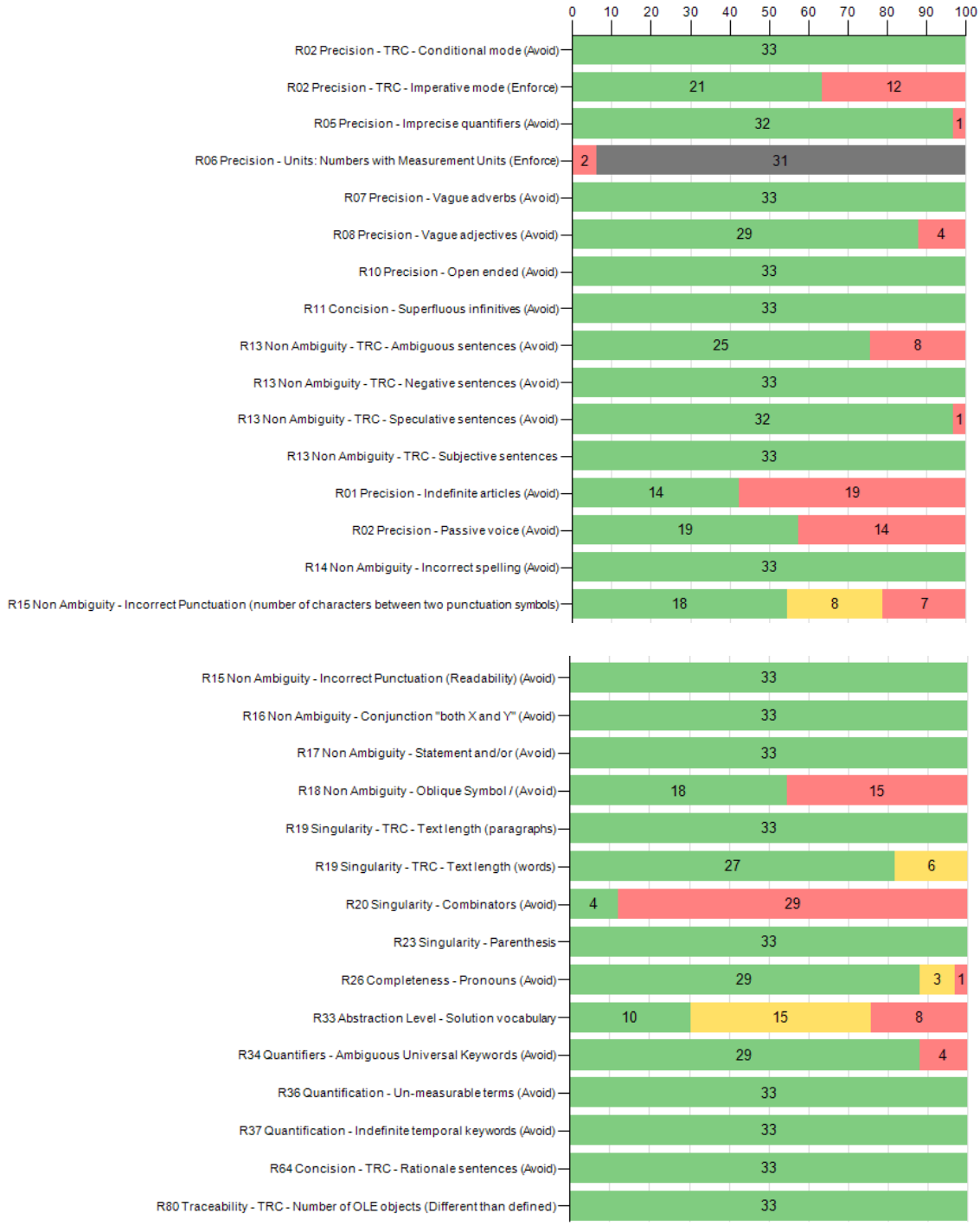


Figura 39. Comportamiento de las métricas en base a los requisitos seleccionados del estándar.

Con este informe podemos concluir que el error más frecuente se produce en la métrica **R20 Singularity – Combinators**. Esta métrica se corresponde al uso abusivo de preposiciones o expresiones de adicción que hacen que el requisito sea demasiado extenso y pueda contener varios requisitos en uno solo. Muchas expresiones de adicción en un solo requisito pueden crear complejidad y ambigüedad a la hora de su interpretación. Las instancias que aparecen son: “And”, “Whether”, “As well as”, “or”.

Otros errores, menos frecuentes que el anterior, que podemos observar se encuentran en la métrica **R02 Precision – Passive Voice** y **R01 Precision - Indefinite articles**.

La primera métrica se corresponde con evitar el uso de la forma verbal pasiva. El uso de la forma pasiva puede crear ambigüedad puesto que esta forma verbal no se deja connotado el sujeto de la acción por lo que queda en duda sobre quien va llevar a cabo ese requisito. Podemos observar como en el 45% de los casos no aparece, pero en el 55% restante sí.

La segunda métrica corresponde con el uso de artículos indefinidos. El uso del artículo indefinido "a" en lugar del artículo definido "the", generan ambigüedad en el requisito al a hora de su interpretación. En este caso podemos observar cómo el 40% de los casos no aparecen, pero en el 60% restante sí.

Los errores menos frecuentes aparecen en las métricas:

- **R02 Precision – TRC – Conditional mode (Avoid):** Esta métrica identifica requisitos que no expresan asertividad. Si un requisito es obligatorio o no, no se expresará en modo condicional.
- **R07 - Vague Adverbs (Avoid):** Esta métrica controla la existencia de adverbios vagos en la declaración de requisitos. Hay que evitar los adverbios vagos, algunos ejemplos son: “usually”, “approximately”, “sufficiently”, “typically”.
- **R10 Precision - Open ended (Avoid):** Esta métrica se cerciora de que no existan clausulas abiertas que puedan conducir al error. Ejemplo de cláusulas abiertas serían etc., sucesivamente…
- **R11 Concision - Superfluous infinitives (Avoid):** Esta métrica controla si un requisito tiene más verbos de lo necesario para describir una acción básica.
- **R13 Non-Ambiguity - TRC - Subjective sentences:** Esta métrica evita el uso de oraciones subjetivas en las que el autor pueda dar su opinión o generar duda.
- **R13 Non Ambiguity - TRC - Negative sentences (Avoid):** Esta métrica identifica requisitos ambiguos al analizar las oraciones negativas en la misma declaración de requisitos. Tener varias expresiones negativas puede generar confusión.
- **R14 Non Ambiguity - Incorrect spelling (Avoid):** Esta métrica corrige los errores ortográficos. Los errores ortográficos pueden causar confusión.
- **R19 Singularity - TRC - Text length (paragraphs)**: Esta métrica identifica los requisitos que están estructurados en más de un párrafo para evitar la sobre especificación, varias necesidades en el mismo requisito o incluso información inútil.

- **R36 Quantification - Un-measurable terms (Avoid):** Esta métrica verifica la existencia de términos no medibles en la declaración de requisitos.
- **R37 Quantification - Indefinite temporal keywords (Avoid):** Esta métrica verifica la existencia de palabras clave temporales indefinidas en la declaración de requisitos. Ejemplo de palabras claves temporales son: a veces, siempre…
- **R64 Concision - TRC - Rationale sentences (Avoid):** Esta métrica identifica las oraciones lógicas en la declaración de requisitos que pueden llevar a la confusión, como "a fin de" o "por eso".

- **R80 Traceability - TRC - Number of OLE objects (Different than defined):** Esta métrica identifica el número de objetos OLE y analiza si los números incluidos en el requisito son diferentes a los definidos en la calidad

5.- CONCLUSION:

5.1.- Conclusiones:

En este punto se analizan las conclusiones obtenidas de la realización de este trabajo. Se tratan tanto desde el punto de vista personal como desde el punto de vista objetivo de la consecución de los objetivos propuestos.

Desde el punto de vista personal, este trabajo ha servido, por un lado, para conocer la problemática existente, en relación con la ambigüedad y complejidad que se da lugar en los estándares de seguridad. La ambigüedad y complejidad existente en los estándares de seguridad dificulta la comprensión, por parte de los proveedores, a la hora de llevar a cabo la implementación de los sistemas críticos de seguridad.

Esta falta de comprensión puede conducir a riesgos de certificación, ya que un proveedor del sistema puede fallar o malinterpretar algunos criterios y por lo tanto no desarrollar un sistema de manera correcta pudiendo desencadenar en fallos. Estos fallos pueden originar consecuencias desastrosas a terceros dentro del ámbito de los sistemas críticos de seguridad.

Por el otro lado, desde una perspectiva más conceptual, ha servido para aprender a interpretar y desarrollar la ontología de un metamodelo RAF y a utilizar desde cero las herramientas RQS, en concreto KM y RQA, con las que se ha llevado a cabo la aproximación propuesta por este trabajo.

Desde el punto de vista objetivo, se han conseguido los objetivos propuestos al principio de este trabajo. Dichos objetivos eran, y se consiguieron, de la siguiente forma:

- **Definir una aproximación para representar estándares de seguridad en KM**: Este objetivo se ha alcanzado a través del entendimiento conceptual del metamodelo propuesto y a través de la aplicación de la herramienta KM.

- **Determinar la correspondencia entre el contenido de una estándar de seguridad y una ontología con la herramienta KM:** Este objetivo ha sido alcanzado mediante el análisis y síntesis del metamodelo propuesto a través del cual se han identificado los tipos de elementos que deben tenerse en cuenta al tener que demostrar el cumplimiento de los estándares de seguridad, así como las relaciones entre ellos. Dentro de este objetivo se encontraban los siguientes subjetivos:

    - **Comprensión de los estándares**: Dicho objetivo se ha alcanzado mediante el análisis de los estándares seleccionados.
    - **Análisis y representación de la ontología de un estándar**: Este objetivo se ha alcanzado a través de la identificación y relación de

términos de los estándares seleccionados con la creación de la propia ontología a través de la herramienta KM.

- **Lograr manejar a la perfección el paquete de herramientas RQS**: Se ha alcanzado mediante un entendimiento y compresión del conjunto de herramientas a la hora de su aplicación.

- **Aplicar y validar la aproximación en casos reales**: Este objetivo se ha alcanzado a través de la aplicación de la herramienta RQA en la ontología creada por la herramienta KM.

También cabe hacer mención especial al artículo escrito a través de la realización de este trabajo cuyo título es: "Representation of Safety Standards with Semantic Technologies Used in Industrial Environments", el cual se ha publicado en el taller SASSUR 2017 (6th International Workshop on Next Generation of System Assurance Approaches for Safety-Critical Systems).

5.2.- Líneas futuras:

En este trabajo se propone el uso de representaciones estructuradas explícitas de las normas de seguridad para facilitar el cumplimiento de las normas, y las tecnologías semánticas pueden utilizarse para crear dichas representaciones. Sin embargo, es necesario seguir trabajando en el tema y debe vincularse y basarse en las prácticas industriales.

Este trabajo se postula como un preámbulo inicial a la hora de representar los estándares de seguridad con tecnologías semánticas ya utilizadas en entornos industriales.

La propuesta representa un uso novedoso de la herramienta KM y un intento de reducir la brecha entre los beneficios que las tecnologías semánticas pueden permitir y cómo se utilizan en la práctica crítica de la ingeniería de sistemas para fines de seguridad.

La propuesta se encuentra en una fase inicial y es necesario seguir trabajando para desarrollarla plenamente. Se plantea promulgar los usos presentados en la Sección 3, lo que permitirá identificar oportunidades de mejora. Las nuevas capacidades que KM tendrá en el futuro (por ejemplo, bibliotecas de ontologías) también pueden permitir usos adicionales.

Finalmente, el trabajo se está realizando dentro del alcance de AMASS (http://amass-ecsel.eu/), que es un gran proyecto H2020-ECSEL de la industria-academia sobre aseguramiento y certificación de sistemas cibernéticos. Así, se podrá aplicar la propuesta en estudios de casos industriales.

6.- REFERENCIAS:

- De la Vara J., Gómez A., Gallego E., Génova G., Fraga A. (2017). *Representation of Safety Standards with Semantic Technologies Used in Industrial Environments.*
- De la Vara J., Ruiz A., Attwood K., Espinoza H., Panesar-Walawege R., López A., del Río O., Kelly T. (2016). *Model-based specification of safety compliance needs for critical systems: A holistic generic metamodel.*
- Gallina B., Pitchai K., Lundqvist K. (2014). *S-TunExSPEM: Hacia una extensión de SPEM 2.0 a los procesos orientados a la seguridad ajustables de modelo y de intercambio.*
- Gallina B., Szatmari Z. (2014). *Ontology-Based Identification of Commonalities and Variabilities Among Safety Processes.*
- Gribov V., Voos H. (2013). *Safety Oriented Software Engineering Process for Autonomous Robots.*
- Hatcliff J., Wassyng A., Kelly T., Comar C., Jones P. (2014). *Certifiably Safe Software-Dependent Systems: Challenges and Directions.*
- Henning J., Kohler S., Koster F. (2016). *Towards a Safer Development of Driver Assistance Systems by Applying Requirements-Based Methods.*
- Hulin B., Kaindl H., Rathfux T., Popp R., Arnautovic E., Becker R. (2016). *Towards a Common Safety Ontology for automobiles and Railway vehicles.*
- Knight J. (2002). *Safety Critical Systems: Challenges and Directions.*
- Kuschnerusy D., Brunsy F., Bilgic A., Muschy T. (2014). *A UML Profile for the Development of IEC 61508 Compliant Embedded Software.*
- Panesar-Walawege R., Sabetzadeh M., Briand L. (2011). *Using Model-Driven Engineering for Managing Safety Evidence: Challenges, Vision and Experience.*
- Rupano, C. Buckl, L. Fiege, M. Armbruster, A. Knoll, G. Spiegelberg. (2014). *Employing early model-based safety evaluation to iteratively derive EE architecture design.*
- Stallbaum H., Rzepka M. (2010). *Toward DO-178B-compliant Test Models.*
- Wu J., Yue T., Shaukat A., Huihui Z. (2013). *Ensuring Safety of Avionics Software at the Architecture Design Level_ An Industrial Case Study.*
- Yaping L., Van den Brand M., Engelen L., Favaro J., Klabbers M., Sartori G. (2013). *Extracting Models from ISO 26262 for Reusable Safety Assurance.*

ANEXO

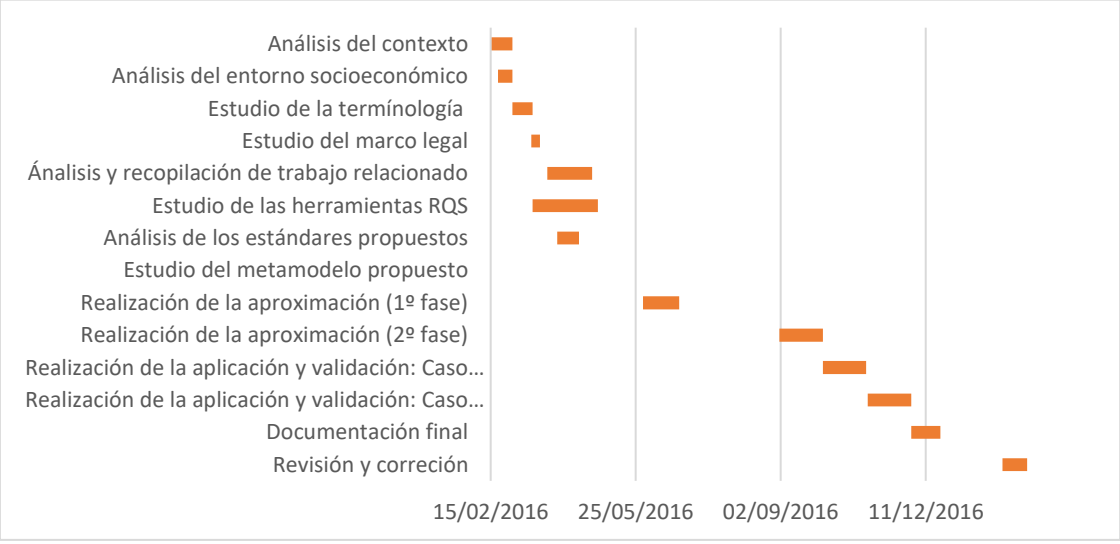
ANEXO 1.- Diagrama de Gantt:

En este apartado se presenta el diagrama de Gantt asociado al proyecto. En este diagrama se representan todas las tareas que se han llevado a cabo para la elaboración de este trabajo.

Todas las tareas tienen una fecha de inicio y fin y duración correspondiente, como podemos observar a continuación en la siguiente tabla:

Actividad	Inicio	Duración (días)	Fin
Análisis del contexto	15/02/2016	15	30/02/2016
Análisis del entorno socioeconómico	20/02/2016	10	02/03/2016
Estudio de la terminología	01/03/2016	14	14/03/2016
Estudio del marco legal	14/03/2016	6	20/03/2016
Análisis y recopilación de trabajo relacionado	25/03/2016	31	25/04/2016
Estudio de las herramientas RQS	15/03/2016	45	31/04/2016
Análisis de los estándares propuestos	01/04/2016	15	15/04/2016
Estudio del metamodelo propuesto	31/04/2016	20	20/05/2016
Realización de la aproximación (1º fase)	30/05/2016	25	25/06/2016
Realización de la aproximación (2º fase)	01/09/2016	30	31/09/2016
Realización de la aplicación y validación: Caso DO-178C	01/10/2016	30	31/10/2016
Realización de la aplicación y validación: Caso EN-50128	01/11/2016	30	30/11/2016
Documentación final	01/12/2016	20	20/01/2016
Revisión y corrección	02/02/2017	17	19/02/2017

A continuación, se presenta el gráfico de Gantt:



ANEXO 2.- Presupuesto:

En este punto se presenta el coste que ha sido necesario para llevar a cabo este proyecto. Los costes imputados se dividen en dos tipos:

- Coste de personal:

  Para llevar a cabo este proyecto ha sido necesario la participación de dos trabajadores: un ingeniero junior y un ingeniero senior. Los salarios establecidos han sido los siguientes:

Trabajador	Sueldo bruto anual	Coste por hora	Horas dedicadas	Coste final
Ingeniero Junior	20.000,00 €	9 €	600	5.400,00 €
Ingeniero Senior	30.000,00 €	15 €	150	2.250,00 €
			TOTAL	**7.650,00 €**

- Coste material:

  Los costes imputados al proyecto en concepto de material con sus respectivas amortizaciones han sido los siguientes:

Objeto material	Coste total	Amortización (meses)	Tiempo de uso (meses)	Coste final asociado
Ordenador ASUS	700 €	60	12	140,00 €
Paquete Office	150,00 €	36	12	50,00 €
Herramientas RQS	- €	N/A	N/A	- €
Material de oficina	N/A	N/A	N/A	50,00 €
			TOTAL	**240,00 €**

Por lo que la suma correspondiente al coste total del proyecto ha sido de: **7650€ + 240€ = 7890€.**

ANEXO 3.- Resumen trabajo en inglés

Executive summary:

Safety-critical systems are those systems whose failure can cause loss of life, significant material damage or damage to the environment.

Critical systems must comply with safety standards as a way to ensure that they do not cause undue risks to people, property or the environment. A safety standard is a document that includes a set of best practices, agreed by a consortium of companies and professionals, for the development and assurance of safety critical systems.

Compliance with safety standards is a very demanding activity, since standards can consist of hundreds of pages and professionals usually have to demonstrate compliance with thousands of safety criteria.

These documents are often long, ambiguous, and difficult to understand, so several experts recommend their explicit and structured representation to facilitate the understanding and application of these standards.

Since the realization of these representations can be complex, it is advisable to use tools that support it.

The objective of this TFG is to define an approach to represent safety standards in RQS, a set of ontology-based requirements engineering tools that is currently used in industry to represent, for example, the requirements and structure of systems.

The approach will also use as a basis the most recent existing proposals for the modeling of safety standards.

Context:

Safety-critical systems are those systems whose failure can cause loss of life, significant material damage or damage to the environment. (Knight, 2015).

There are many well-known examples in areas of application such as medical devices, aircraft flight control, weapons and nuclear systems. (Ibid).

Many modern information systems are becoming critical in a general sense, by the financial losses and even lives that can result from their failures. Future safety-critical systems will be more common and more powerful. (Ibid).

From a software perspective, the development of safety-critical security systems in the numbers required and with adequate reliability will require significant advances in areas such as specification, architecture, verification and process. (Ibid).

The visible problems that have also arisen in the area of security of information systems suggest that security is also a major challenge. (Ibid).

Certification is an important prerequisite for most safety-critical systems before they can be put into operation. (Panesar-Walawege, Sabetzadeh and Briand, 2011).

During certification, suppliers of safety-critical security systems often have to present a coherent set of evidence demonstrating that the systems developed are safe for the operation. (Ibid).

Regardless of the approach to certification adopted (based on the process or the product), the collection of appropriate evidence during development is critical for successful certification. (Ibid).

Currently, both safety-critical security systems providers and certification bodies face several challenges in relation to the collection and enforcement of safety tests. (Ibid).

In particular, they find it difficult to interpret the evidence requirements imposed by safety regulations within the scope of application. There is little help to record, consult and report the evidence in a structured manner. And there is a general lack of guidelines on how the evidence collected supports the safety objectives. (Ibid).

Motivation:

Most safety-critical systems must comply with safety standards as a way to ensure that they do not cause undue risk. Examples of these standards include IEC 61508 for a wide range of industries, DO-178C in avionics, EN 50128 in the railroad sector, and ISO 26262 in the automotive industry. (José Luis de la Vara, Álvaro Gómez, Elena Gallego, Gonzalo Génova and Anabel Fraga, 2017).

Safety standards are typically large textual documents that consist of hundreds of pages and define thousands of criteria for compliance. (Ibid).

The resulting complexity can make it difficult to understand a standard. Ambiguity and contradictions are also common in their text, and suppliers in the sector have recognized problems in the understanding and application of standards. (Ibid).

This can lead to certification risks, since a system provider may fail or misinterpret some criteria and therefore not develop a compatible system. (Ibid).

As a solution, this TFG proposes the use of structured representations of safety standards to help providers of critical systems to understand and follow them. (Ibid).

These representations have most often been UML-based models, such as a class diagram or UML profile. However, representations can also be developed with semantic technologies, for example, as an ontology that includes the main concepts and relationships between the concepts of a safety standard.

This TFG aims to address these problems by investigating how the KM (Knowledge Manager) tool can be used to represent safety standards and then exploit the resulting representation.

KM is a tool used in industrial environments for the engineering of critical systems to represent knowledge of the domain with ontologies. These ontologies cover several aspects, from system terminology to system specification patterns, and can be used for different purposes, for example, system specification, quality analysis of the system artefact and reuse of system information.

The use of KM in practice focuses on specific characteristics of the system, for example, structure of the system, but we argue that such use can be extended to support compliance with safety standards.

Objective:

The objective of this TFG is to define an approach to represent safety standards in KM, a requirements engineering tool, belonging to the RQS toolkit, based on ontologies that is currently used in industry to represent, for example, the requirements and the structure of systems. The approach will also use the most recent existing proposals for the modeling of safety standards.

This will ensure that the analysis of safety standards for subsequent use in the creation of critical safety systems becomes a simpler task and not as complex as it is today.

Within this general objective, the following specific objectives can be differentiated:

- Determine the correspondence between the content of a safety standard and an ontology with the KM tool.
    - This objective will in turn require:
        - Understanding of the standards.
        - Analysis and representation of the ontology of a standard.
        - Manage the RQS tool package perfectly.
- Apply and validate the approach in real cases.

RQS:

Requirements Quality Suite (RQS) is a set of tools that support planning, personalization, measurement, control and management of work products, especially their requirements specifications, thus improving their quality. (The Reuse Company, 2016).

RQS evaluates how closely requirements match the quality characteristics described in references known as IEEE Std. 830, IEEE 29148 or ESA PSS-05 and others. (Ibid).

These characteristics are evaluated by means of a target set and easy to measure (also known as rules or indicators) that can be easily customized at the level of maturity of your organizational processes or project and that always corresponds to the set of guidelines or lists of defined verification and mandate of your organization. (Ibid).

RQS can analyze the following languages: English, French, German, Swedish and Spanish, but can be easily adapted for any other language when necessary. (Ibid).

The tools included in the suite are as follows:

• Requirements Quality Analyzer (RQA): allows the project to measure the quality of its requirements with the guides and checklists of the organization. It also helps the Quality Control (QC) team during verification activities to assess compliance.

• Requirements authoring tool (RAT): helps the team in charge of drafting the specifications of the requirements during this critical activity: checking accuracy, checking coherence, reusing requirements ... all on the fly and in real time. In addition, writing assistance is based on patterns (aka boilerplates or declaration level templates) that promotes consistency and improves reuse.

• Knowledge Manager (KM): for the management of controlled vocabularies, semantics, ontologies and patterns used during the analysis of requirements.

• RQS (Requirements Quality Suite) Server: This server can be connected to IBM Excel, XML, or DOORS files. The RQS tool was created by the REUSE company to efficiently handle the quality of the requirements. Apart from the products explained above, the application includes a series of tools as well as defining a PDCA cycle to provide the necessary methodology for the process. (Ibid)

Approach:

In this section the methodology to apply the approach is explained. The tool that has been used to carry out the approach has been Knowledge Manager (KM).

The proposed approach consists of two main activities: KM configuration and information specification of a standard, which in turn is divided into two sub-activities: specification of the terminology of a standard and specification of the conceptual model of a safety standard.

Each activity consists of several steps, as explained below.

1st Phase: Configuration KM:

This activity is necessary to adapt the default use of KM to represent safety standards, that is, certain aspects of KM must be configured so that a user can create an appropriate representation according to the generic holistic metamodel.

The configuration focuses on the semantic aspects of the rules that must be included in the representation. These aspects are specific to safety standards but are independent of the specific standard to be represented. Two tasks must be performed.

1.1.- Specification of semantic groups:

New clusters must be added to the conceptual model layer in order to indicate the type of information that a term represents.

First, a cluster with the name of the safety standard to be represented is required to specify later that a term falls within the scope of the standard.

Second, the cluster types belonging to the metamodel will be introduced. These clusters will have the purpose of modeling the different elements of an RAF.

Its representation in the metamodel is given by rectangles. The types of clusters are the following:

- Reference Artefact.
- Reference Activity.
- Reference Role.
- Reference Artefact Atribute.
- Reference Technique.

-Commands to introduce clusters in KM:

1.- Right click inside the term table in the terminology section.

2.- Next, a series of options will be displayed. The option add new term will be clicked.

3.- Right click on the blank space belonging to the Cluster section (s) and click on add new cluster.

4.- Right click inside the “clusters” table and click on add new cluster.

5.- Enter name of the cluster and click on accept. In this way, the enunciated clusters will be introduced one by one.

-Note: The cluster type Reference Requirement will not be represented by this tool because it has no scope to extend its terms. This type of cluster will be treated later with another tool that we will cite in the next point.

1.2.- Specification of types of relationships:

KM also supports the specification of types of relationship between terms. To represent a safety standard, you must create a relationship type for each association in the metamodel between the metaclasses.

The types of relationships belonging to the model among the different types of clusters will be introduced. These relations will have the purpose of associating the different elements of an RAF. Its representation in the metamodel is given by the lines that join the rectangles. Within the tool, relationships are called views you will have to create a view for each type of relationship in the model.

The types of relationships are the following:

- Taxonomía (Ya definida por la herramienta).
- PBS (Ya definida por la herramienta).
- Input.
- Output.
- Technique.
- Predecessor.
- Participant.
- Reference Artefact RelationShip.

-*Commands to introduce relationships:*

1.- Click on the section of the Conceptual Model.
2.- Click on the Relationship Types icon.
3.- Right click on the Types of relationship table and click on add new relationship type.
4.- Introduce name of the relationship and roles. Roles are the words that appear at the ends of the line joining the clusters. In this way all the relationships of the model will be introduced one by one.

2nd Phase: Specification of the information of a standard:

This activity results in the specific representation of a given safety standard. You can distinguish two tasks. Normally, tasks will be executed iteratively to incrementally represent a safety standard.

2.1 Specification of the terminology of a standard:

In this step the text will be analyzed in search of specific terms of the standard. Once identified, they will be modeled in one cluster or another, depending on their purpose within the standard.

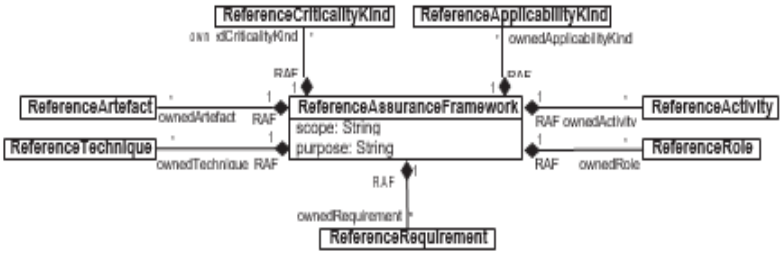


Figure 4. Metamodel clusters

This task has two main aspects to deal with:

1. Glossary terms: First, the terms of the glossary with their respective acronyms will be analyzed and classified in their corresponding cluster. To do this, two records will be made, first that of the acronym and then the full term.

   Each time a term is added, it is necessary to (1) specify its syntactic category (for example, noun or acronym) and (2) associate it with the clusters to which it belongs, in this case in the one of the standard itself and in the cluster corresponding within the standard according to its semantic meaning.

2. Terms of the standard: Next, the text of the standard should be analyzed to identify the terms that correspond to the Reference Artefact, the Reference Artefact Attribute, the Reference Activity, the Reference Role or the Reference Technique. Each time a term is identified, it is added to the terminology and the semantic cluster to which it belongs is associated to the cluster of the standard. Once the terms have been identified, the next step is to model them in a metamodel cluster.

Each time a term is identified, it is added to the terminology and the semantic cluster to which it belongs is associated to the cluster of the standard. Once the terms have been identified, the next step is to model them in a metamodel cluster.

- Commands to add terms:

1.- Click on the Terminology tool section.
2.- Right click inside the term table in the terminology section.
3.- Next, a series of options will be displayed. The option add new term will be clicked.
4.- Fill in the name and syntactic label fields.
5.- Select the clusters to which it belongs. Each term will have to be classified by two clusters:
   1.- Cluster to which the term belongs within the metamodel. Example: Reference Artefact.
   2.- Cluster own standard to which the term belongs. Example: DO- 178.

This process will be repeated successively until there are no more terms left in the standard to analyze.

2.2 Modeling relationships between terms:

Once all the relevant terms have been entered and classified, the relationships between them can be specified in the Conceptual model. These relationships will be classified according to the types of relationship available in KM, both the default ones and those created during the KM configuration.

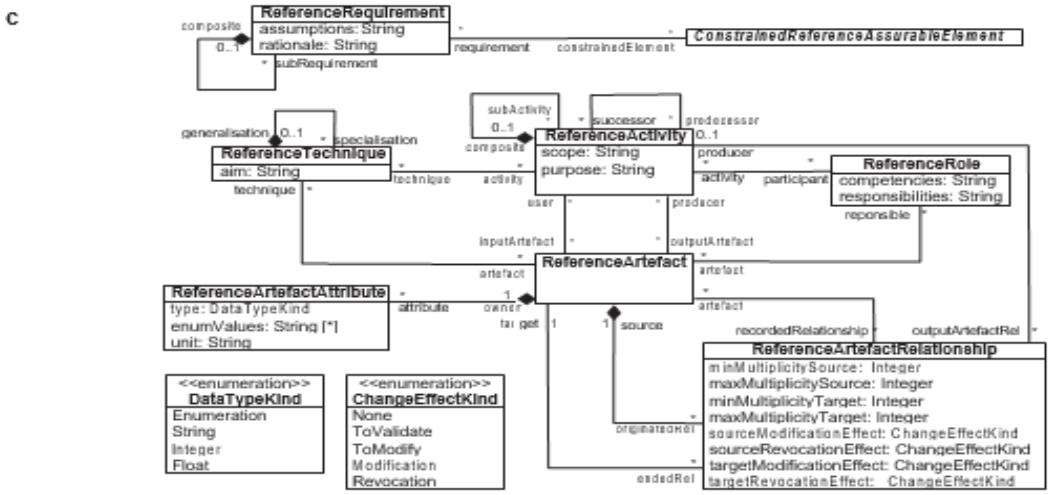


Figure. Metamodel RAF associations.

A user must conform to the holistic generic metamodel by specifying relationships, that is, only terms that correspond to the ends of a given association in the metamodel must be related.

The relationships to be carried out will be those mentioned in the first section. Then, they will be described one by one:

- Taxonomy: This relationship will be used when establishing a taxonomy between terms.
- Equivalence: This relationship will be used when matching acronyms with their term.
- PBS: This relationship will be used at the time of the decomposition of an artefact or a process, that is to say, what are the components by which an artefact is formed or what are the sub-activities by which it is formed of a process.
- Input: This relation will be used to represent the input of artefacts to the different processes of the standard for the subsequent creation of an output.
- Output: This relationship will be used to represent which artefacts are created by the different processes of the standard.
- Technique: This relationship will be used to model a specific technique within a reference activity. It can also be applied to an artefact.
- Predecessor: This relationship serves to model the order of the processes of the standard, that is, which is what happens before.

• Participant: This relationship serves to assign the artefact or process to a role. • Reference Artefact Relationship: This relationship is used to carry out relationships between artefacts. It is usually defined by the verb that accompanies both artefacts in the standard. Ex.: conform to, verifies, base on ...

The user must also decide whether the relationships between Reference Artefacts should be specified as specializations, such as compositions or according to the type of Artefact Reference ratio.

It is possible to define specializations of this type of relationship if a user decides it, for example, because it is a recurrent Reference Artefact Ratio. For example, it is common for artifacts to have to be "conforming" to some plan or standard.

Finally, it may also be necessary to specify specialization and equivalence relationships between terms; For example, 'MC / DC'; and 'Modified Condition / Decision Coverage' are equivalent for DO-178C.

The process of establishing relationships will be divided into three parts:

1. Acronym Modeling: In this initial part, the relationships between the acronyms and their respective glossary terms will be established. The relationship to be used will be "Equivalence".

2. Process Modeling: In this part the relations of the terms related to the process will be established. The participating terms will be the Reference Activity and Reference Artefact. The relationships to be carried out will be only output, input, subactivity (PBS) and predecessor. The objective of this part consists basically in modeling which are the processes that are preceded, which are the activities by which they are formed, and which are their inputs and outputs.

3. Artifact Modeling: In this part the relationships between the terms related to the artefacts will be established. The participating terms will be the Reference Artefact. The relationships to be carried out will be Taxonomy, PBS and Reference Artefact Relationship.

The objective of this part is to model the relationships between the different artefacts of the standard and its composition.

-Commands to establish relationships:

1.- Click on the section of the "Conceptual Model".
2.- Click on the type of view that we want to introduce.
3.- Right click on the blank table by default and a series of options will appear.
4.- Within the options, if we want to create a parent term of the relationship we will click on the option "add new term or relation" and if we want to add a term to a relation we will right click on the parent term and then on "add a new son" This process will be repeated successively until there are no more relationships in the standard to analyze and identify

Conclusions:

In this point the conclusions obtained from the realization of this work are analyzed. They are treated both from the personal point of view and from the objective point of view of the achievement of the proposed objectives.

From the personal point of view, this work has served, on the one hand, to know the existing problematic, in relation to the ambiguity and complexity that occurs in the safety standards. The ambiguity and complexity of safety standards makes it difficult for providers to understand the implementation of critical safety-critical systems.

This lack of understanding can lead to certification risks, since a supplier of the system can fail or misinterpret some criteria and therefore not develop a system correctly and can trigger failures. These failures can cause disastrous consequences to third parties within the scope of critical safety-critical systems.

On the other hand, from a more conceptual perspective, it has served to learn to interpret and develop the ontology of a RAF metamodel and to use the RQS tools, specifically KM and RQA, from scratch, with which the proposed approach has been carried out. for this job.

From the objective point of view, the objectives proposed at the beginning of this work have been achieved. These objectives were, and were achieved, in the following way:

• Define an approach to represent safety standards in KM: This objective has been achieved through the conceptual understanding of the proposed metamodel and through the application of the KM tool.

• Determine the correspondence between the content of a safety standard and an ontology with the KM tool: This objective has been achieved through the analysis and synthesis of the proposed metamodel through which the types of elements that must be taken into account when having to demonstrate compliance with safety standards, as well as the relationships between them.

Within this objective were the following subjective:

- o Understanding of the standards: This objective has been reached through the analysis of the selected standards.
- o Analysis and representation of the ontology of a standard: This objective has been achieved through the identification and relation of terms of the selected standards with the creation of the ontology itself through the KM tool.
- o Manage the RQS tool package perfectly: It has been achieved through the understanding of the set of tools at the time of its application.

• Apply and validate the approach in real cases: This objective has been achieved through the application of the RQA tool in the ontology created by the KM tool.

I would also like to make special mention to the article written through the realization of this work whose title is: “Representation of Safety Standards with Semantic Technologies Used in Industrial Environments”, which has been published at the SASSUR 2017 workshop (6th International Workshop on Next Generation of System Assurance Approaches for Safety-Critical Systems).

ANEXO 4.- Ejemplo de tabla importación RQA:

ID	CUSTOM_ID	Short text	Description	Author	Creation_date	Last_Update	QualityLevel	NumericQuality	QualityDate	QualitySummary	QualityLevel_1	NumericQuality_1	QualityDate_1	QualitySummary_1
00001	5.0-1	SW Dev Proc parts	The software development processes are: Software requirements process, Software design process, Software coding process, Integration process.	Alvaro	20-abr	20-abr					N/A	0	01/01/1900	N/A
00002	5.0-2	Produced levels	Software development processes produce one or more levels of software requirements	Alvaro	20-abr	20-abr					N/A	0	01/01/1900	N/A
00003	5.0-3	HLR	High-level requirements are produced directly through analysis of system requirements and system architecture	Alvaro	20-abr	20-abr					N/A	0	01/01/1900	N/A
00004	5.0-4	HLR development	Usually, these high-level requirements are further developed during the software design process, thus producing one or more successive, lower levels of requirements.	Alvaro	20-abr	20-abr					N/A	0	01/01/1900	N/A
00005	5.0-5	Source Code	However, if Source Code is generated directly from high-level requirements, then the high-level requirements are also considered low-level requirements and the guidance for low-level requirements also apply.	Alvaro	20-abr	20-abr					N/A	0	01/01/1900	N/A
00006	5.0-6	Developement of software	The development of software architecture involves decisions made about the structure of the software.	Alvaro	20-abr	20-abr					N/A	0	01/01/1900	N/A
00007	5.0-7	SW Des process	During the software design process, the software architecture is defined and low-level requirements are developed.	Alvaro	20-abr	20-abr					N/A	0	01/01/1900	N/A
00008	5.0-8	Low-level requirements	Low-level requirements are software requirements from which Source Code can be directly implemented without further information.	Alvaro	20-abr	20-abr					N/A	0	01/01/1900	N/A
00009	5.0-9	Produced derived requirements	Each software development process may produce derived requirements.	Alvaro	20-abr	20-abr					N/A	0	01/01/1900	N/A
00010	5.0-10	HLR may incude	High-level requirements may include derived requirements, and low-level requirements may include derived requirements.	Alvaro	20-abr	20-abr					N/A	0	01/01/1900	N/A